Novelty Heuristics, Multi-Queue Search, and Portfolios for Numeric Planning

Dillon Z. Chen[1,2], **Sylvie Thiébaux**[1,2]

[1]LAAS-CNRS, Université de Toulouse
[2]Australian National University
{dillon.chen, sylvie.thiebaux}@laas.fr

Abstract

Heuristic search is a powerful approach for solving planning problems and numeric planning is no exception. In this paper, we boost the performance of heuristic search for numeric planning with various powerful techniques orthogonal to improving heuristic informedness: numeric novelty heuristics, the Manhattan distance heuristic, and exploring the use of multi-queue search and portfolios for combining heuristics.

1 Introduction

Numeric planning is an expressive extension of classical planning where states are able to exhibit real-valued variables. It was formalised in PDDL2.1 (Fox and Long 2003) and is undecidable in the general case (Helmert 2002). However, it is PSPACE-complete when variables are integer and bounded, and there also exist compilations from numeric to classical planning with certain features that preserve plan length (Gigante and Scala 2023). Similarly to classical planning, the two main approaches for solving numeric planning problems in the literature consist in either heuristic search or compilation into a sequence of satisfiability problems for constraint-based solvers. A recent SAT Modulo Theory (SMT) approach, PATTY, has been proposed (Cardellini, Giunchiglia, and Maratea 2024) which provably encodes fewer variables and constraints than previous constraint-based approaches (Scala et al. 2016b; Bofill, Espasa, and Villaret 2017). Compared to heuristic search, it provides better performance on highly numeric planning problems which do not require long plans.

On the other hand, the state of the art for heuristic search in numeric planning still consists of a search with a single heuristic derived from a relaxation of the problem. Such heuristics include the numeric extensions of classical planning heuristics: metric FF (Hoffmann 2002, 2003), LP-based heuristics for computing tighter bounds on the relaxed planning graph (Coles et al. 2008), interval-based, subgoaling, and set-additive derivations of h^{add} (Scala, Haslum, and Thiébaux 2016; Scala et al. 2016a, 2020a,b), landmarks (Scala et al. 2017), admissible IP-based heuristics (Piacentini et al. 2018), LM-cut and operator counting heuristics (Kuroiwa et al. 2021, 2022; Kuroiwa, Shleyfman, and

Beck 2023). Alternatively, Illanes and McIlraith (2017) use a partial policy computed from an abstraction of numeric planning into classical planning for search guidance. Other techniques orthogonal to defining new heuristics for numeric planning also exist, such as abstracting linear numeric problems into simple numeric problems, enabling the use of simple numeric heuristics (Li et al. 2018), and computing state space symmetries with the numeric problem description graph (Shleyfman, Kuroiwa, and Beck 2023).

In this paper, we boost the performance of heuristic search for numeric planners in several orthogonal directions by porting search techniques from classical planning. This includes (1) formalising novelty heuristics for numeric planning, (2) constructing the simple but effective Manhattan distance heuristic, (3) extending the ENHSP planner (Scala et al. 2020b) with multi-queue search, and (4) building naive portfolios. We provide a systematic set of experiments that demonstrate the effectiveness of these approaches on the Numeric Track of the 2023 International Planning Competition (NT-IPC) (Arxer and Scala 2023) and highlight the complementary nature of heuristic search and constraint-based approaches for numeric planning.

2 Background

A numeric planning task (Fox and Long 2003) is given by a tuple $\Pi = \langle X_p, X_n, A, s_0, G \rangle$ where X_p is a finite set of Boolean variables with domains $\{\top, \bot\}$ and X_n is a finite set of numeric variables with domains $\mathbb{R}$. Let $X = X_p \cup X_n$ denote the set of state variables. Let S denote the set of possibly infinitely many states in Π, where a state is a total assignment to Boolean and numeric variables. Let $s_0 \in S$ be the initial state. A propositional condition is a positive literal, and a numeric condition has the form $\xi \unrhd 0$ where ξ is an arithmetic expression over numeric variables and $\unrhd \in \{\geq, >, =\}$. Let $[x]^s$ and $[\xi]^s$ denote the value of x and ξ in s, respectively. The goal condition G is a set of propositional and numeric conditions, and we say that a state s satisfies G if s satisfies all conditions in G. An action a consists of preconditions and effects. Action preconditions $\mathrm{pre}(a)$ are sets of propositional and numeric conditions, and action effects $\mathrm{eff}(a)$ assign Boolean values to Boolean variables, and assign the value of a numeric variable using an arithmetic expression. An action a is applicable in a state s if s satisfies $\mathrm{pre}(a)$, in which case its successor $a(s)$ is the

state where the effects $\mathrm{eff}(a)$ are applied to the variables in state s. If a is not applicable in s, we set $a(s) = s_\perp \notin S$. A plan for a numeric planning task is a sequence of actions $a_1, \ldots, a_n$ such that $s_i = a_i(s_{i-1}) \neq s_\perp$ for all $1 \leq i \leq n$ and s_n satisfies G. A numeric planning task is solvable if there exists a plan for it.

3 Boosting Search for Numeric Planning

In this section, we outline various techniques to boost the performance of satisficing numeric planning solvers. Firstly, we extend and combine novelty heuristics into a single framework for numeric planning. We then introduce a simple extension of the goal count heuristic for numeric planning which outperforms more sophisticated numeric planning heuristics for single queue search. Lastly, we make use of multi-queue search and portfolios for numeric planning.

3.1 Numeric Novelty Heuristics

Let $S^{<\mathbb{N}}$ denote the set of all finite sequences of states in a numeric planning task, and $\langle T \rangle$ a finite sequence of states. We write T_k for the state of index k in $\langle T \rangle$, and $\langle T \rangle . s$ for the sequence obtained by appending state s at the end of $\langle T \rangle$. Moreover, let $\langle T \rangle [: s]$ stand for the substring of $\langle T \rangle$ that ends in the element before s, and $\langle T_{=h(s)} \rangle$ for the substring of $\langle T \rangle$ consisting of the states s' such that $h(s') = h(s)$, where h is a heuristic function. We also assume an enumeration $x_1, \ldots, x_N$ of variables in X, where $N = |X|$.

Definition 3.1 (Novelty Feature). A *novelty feature* is a function $f : S^{<\mathbb{N}} \times S \to (\mathbb{R} \cup \{\perp\})^N$.

We provide two examples of novelty features for numeric planning. The first example is the *assignment feature* A, which has been explored in the context of control systems (Ramírez et al. 2018). The function maps a state to a vector of its assignments. More specifically, the i-th element of $\mathrm{A}(\langle T \rangle, s)$ is the value of x_i in s if x_i is a numeric variable and otherwise if $x_i = \top$ the element is 1 and else $\perp$. Note that the function is agnostic to the $\langle T \rangle$ input.

The next example is the *boundary extension encoding feature* (Teichteil-Königsbuch, Ramírez, and Lipovetzky 2020) B, which maps a state to a vector whose component for a particular variable reflects the number of times this variable reached a new extreme value in the past before reaching its current value. More formally, for a variable x_i and state s such that $[x_i]^s > [x_i]^{s_0}$, the novelty feature $I(\langle T \rangle, s, i)$ is the number of times x_i is strictly greater than its value in all preceding states in T before reaching a state s_j where $[x_i]^{s_j} \geq [x_i]^s$. That is

$$I(\langle T \rangle, s, i) = \sum_{k=1}^{J_I} \left(\bigwedge_{l=1}^{k-1} [x_i]^{T_k} > [x_i]^{T_l} \right), \quad (1)$$

where $J_I \leq |\langle T \rangle|$ is the smallest index j such that $[x_i]^{T_j} \geq [x_i]^s$, and the boolean expression evaluates to 1 or 0. The case where $[x_i]^s < [x_i]^{s_0}$ is similar, but we now look at decreasing values of x_i with

$$D(\langle T \rangle, s, i) = \sum_{k=1}^{J_D} \left(\bigwedge_{l=1}^{k-1} [x_i]^{T_k} < [x_i]^{T_l} \right), \quad (2)$$

where $J_D \leq |\langle T \rangle|$ is the smallest index j such that $[x_i]^{T_j} \leq [x_i]^s$. We can then define the boundary extension encoding feature B on its individual component variables

$$\mathrm{B}(\langle T \rangle, s)_i = \begin{cases} 0, & \text{if } [x_i]^s = [x_i]^{s_0} \\ I(\langle T \rangle . s, s, i), & \text{if } [x_i]^s > [x_i]^{s_0} \\ -D(\langle T \rangle . s, s, i), & \text{if } [x_i]^s < [x_i]^{s_0}. \end{cases} \quad (3)$$

Next, we move on to defining a novelty heuristic given a novelty feature and a base heuristic. We note that the definition can be generalised to take as input a tuple of heuristics but we omit this for simplicity of notation.

Definition 3.2 (Novelty Heuristic). Given a novelty feature f and a heuristic h, a *novelty heuristic* is a function $n_f^h : S^{<\mathbb{N}} \times S \to \mathbb{R}$.

We again provide two examples of novelty heuristics. The first is the *partition novelty* (PN) measure heuristic. It was first introduced for solving planning problems in polynomial time with an incomplete search (Lipovetzky and Geffner 2012) and later adapted for heuristic search (Lipovetzky and Geffner 2017; Corrêa and Seipp 2022). Given an integer $k \geq 1$, we extend its definition to numeric planning by the function $^k\mathrm{PN}_f^h$ where $^k\mathrm{PN}_f^h(\langle T \rangle, s)$ is the smallest $n \leq k$ for which there is an n-subset of variable indices $i_1, \ldots, i_n$ such that none of $x_{i_j} = \perp$ in s and $f(\langle T_{=h(s)} \rangle, s)[i_1, \ldots, i_n]$ differs from $f(\langle T_{=h(s)} \rangle [: s'], s')[i_1, \ldots, i_n]$ for all $s' \in \langle T_{=h(s)} \rangle$, and $k + 1$ if no such subset exists. The original partition novelty operates on a tuple of heuristics and it is easy to extend this definition to do so.

Another example of a novelty heuristic is the *quantified both* (QB) novelty function (Katz et al. 2017). The weakness of the partition novelty measure is that its value does not prioritise states with the same partition novelty based on the base heuristic value. For example, two states with heuristic values 5 and 3 and the same partition novelty measure will be treated equally in the search, unless a tie-breaking strategy is used. Furthermore, non-novel states are also treated the same. The quantified both novelty function addresses these issues by making use of the base heuristic value and also distinguishing non-novel states. We extend the quantified both novelty function to the numeric case and also to operate on k-subsets of variables instead of just single facts.

With abuse of notation, we omit $\langle T \rangle$ and the novelty feature f from various helper function arguments. Firstly, we quantify the novelty of subsets of variables by the function

$$N(J, s) = \min_{s' \in T, f(\langle T \rangle, s)_i = f(\langle T \rangle, s')_i, \forall i \in J} h(s') \quad (4)$$

where $\min$ returns ∞ if the set it operates over is empty. The function returns the minimum value of the heuristic over the past states that share the same features as s for the considered variable subset. Then we define $\varphi(J, s) = h(s) < N(J, s)$ and $\psi(J, s) = h(s) > N(J, s)$ where the Boolean output evaluates to 0 or 1. The function φ states that the J values of s are novel if $h(s)$ is strictly lower than the heuristic value of all previously seen states with the same J values. Similarly, ψ denotes non-novelty by noting whether the heuristic value is strictly greater than the minimum heuristic

value of previously seen states with the same J values. Let $\left\{{N \atop n}\right\}$ be the set of n-subsets of variable indices of s and $\left({N \atop n}\right)$ its cardinality. Given an integer $k \geq 1$, we can define the quantified both novelty function ${}^k\mathrm{QB}^h_f$ by

$${}^k\mathrm{QB}^h_f(\langle T \rangle, s) =$$

$$\begin{cases} \sum_{n=1}^{1} \binom{N}{n} - \sum_{J \in \{{N \atop 1}\}} \varphi(J,s), & \text{if } \exists J \in \{{N \atop 1}\}, \varphi(J,s) \\ \quad\vdots \\ \sum_{n=1}^{k} \binom{N}{n} - \sum_{J \in \{{N \atop k}\}} \varphi(J,s), & \text{if } \exists J \in \{{N \atop k}\}, \varphi(J,s) \\ \sum_{n=1}^{k} \binom{N}{n} + \sum_{J \in \{{N \atop k}\}} \psi(J,s), & \text{otherwise,} \end{cases} \quad (5)$$

where cases are tie-broken by priority from the top. Unlike the partition novelty measure, it is not as straightforward to define the quantified both novelty function for tuples of heuristics. This is because Eqn. (4) would return a Pareto set for the min function if multiple heuristic were used. One would then require defining $\varphi(J,s)$ over Pareto sets in which case there are several ways to do so.

3.2 Manhattan Distance Heuristic

The *goal count heuristic* h^{gc} for classical planning is the simplest non-degenerate heuristic which counts the number of achieved goal propositions in a state. It has been extended to numeric planning by counting the number of achieved propositional and numeric goal conditions in the current state. However, we can easily refine the heuristic for numeric conditions $\xi \rhd 0$ by measuring the error of the expression ξ evaluated in the current state. To formally define this more refined goal count heuristic, let $G = G_p \cup G_n$ be the goal of our problem where G_p is the set of propositional conditions and G_n the set of numeric conditions. Let $[l]^s = 1$ if s satisfies the literal l and 0 otherwise. Then, given a goal condition G we define the *Manhattan distance heuristic* h^{md} by

$$h^{md}(s) = \sum_{l \in G_p} [l]^s + \sum_{c = \xi \rhd 0 \in G_n} \gamma(c,s) \quad (6)$$

where the error is $\gamma(c,s) = 0$ if s satisfies c and $|[\xi]^s|$ otherwise. It is possible to further refine h^{md} by dividing the error of each numeric condition $\xi \rhd 0$ by some constant computed from the set of actions relevant to ξ. For example, we could choose the constant to be either the min or max action effect that brings $|\xi|$ closer to 0 if the condition is not yet satisfied.

3.3 Multi-Queue Search

Multi-queue search is an effective method for combining heuristics for satisficing search by maintaining a separate queue in greedy best first search for each heuristic (Helmert 2006; Röger and Helmert 2010). More specifically, given n heuristics, we have n priority queues from which we pop nodes to expand in a round-robin manner. After a node is expanded, the children are evaluated by each heuristic and

Numeric Heuristics							Novelty Heuristics			
h^{gc}	h^{md}	h^{aibr}	h^{add}	h^{radd}	h^{mrp}	h^{mrp}_{+hj}	$h^{add}_{\langle A,PN\rangle}$	$h^{add}_{\langle B,PN\rangle}$	$h^{add}_{\langle A,QB\rangle}$	$h^{add}_{\langle B,QB\rangle}$
117	200	119	183	171	176	217	178	181	215	**236**

Table 1: Total coverage of single queue GBFS with a selection of ENHSP heuristics and h^{add} novelty heuristics. The top three scores are indicated by the cell colour intensity.

inserted into the corresponding priority queues. In our implementation, we assume no node reopening which means states are expanded at most once over all queues, at the cost of potentially lower quality solutions. Naturally, when a novelty heuristic and its base heuristic are used in the same multi-queue search, we can save computing the base heuristic's value twice. We note that multi-queue search is not limited to using heuristic values for determining queues, as we may also construct queues from preferred operators (Richter and Helmert 2009; Richter and Westphal 2010) or total orders of states (Garrett, Kaelbling, and Lozano-Pérez 2016).

3.4 Portfolios

Given that there is no single numeric planner configuration that performs best for all domains, the research community introduced portfolios and automatic configuration selection of planners, both of which combine standalone planners to maximise coverage over a diverse set of domains. Both portfolios and automatic configuration selection methods usually perform well on optimal and satisficing tracks of the International Planning Competitions. Portfolios (Helmert, Röger, and Karpas 2011) statically select a set of planners to run in sequence, each with a set amount of timeout such that the sum of their timeout is equal to the total timeout of the portfolio planner. The partitioning of time resources for planners can be selected from training data or can simply be uniform (Seipp et al. 2012). Similarly to portfolios, one may also just learn to choose a specific planner for a given domain (Fawcett et al. 2011; Katz et al. 2018; Ma et al. 2020). To reduce overfitting to benchmarks and to give a better understanding of the state of numeric planning domains, we opt to only use uniform portfolios in our experiments.

4 Experiments

We evaluate the effectiveness of novelty heuristics and multi-queue search for numeric planning on the Numeric Track of the 2023 International Planning Competition (NT-IPC) (Arxer and Scala 2023). Its benchmark suite consists of 20 domains with 20 problems each. Action effects are limited to either linear or simple effects. Heuristics we consider are goal count (h^{gc}), Manhattan distance (h^{md}), the additive interval-based relaxation (h^{aibr}) (Scala et al. 2016a), the subgoaling additive heuristic with (h^{radd}) and without (h^{add}) redundant constraints (Scala, Haslum, and Thiébaux 2016), and the multi-repetition relaxed plan heuristic (h^{mrp}) with successors restricted to states generated by up-to-jumping actions (h^{mrp}_{+hj}). Given a novelty feature f, a heuristic h,

num. tasks	S GBFS						M GBFS			P GBFS			SMT
	h^{md}	h^{add}	$h^{\mathrm{mrp}+\mathrm{hj}}$	$h^{\mathrm{md}}_{\langle\mathrm{B,QB}\rangle}$	$h^{\mathrm{add}}_{\langle\mathrm{B,QB}\rangle}$	$h^{\mathrm{mrp}}_{\langle\mathrm{B,QB}\rangle+\mathrm{hj}}$	$\mathrm{M}(3h)$	$\mathrm{M}(3n)$	$\mathrm{M}(3h\|3n)$	$\mathrm{P}(3h)$	$\mathrm{P}(3n)$	$\mathrm{P}(3h\|3n)$	PATTY
400	200	183	217	185	236	222	261	244	274	290	292	**315**	262

Table 2: Total coverage of single queue (S), multi-queue (M), and portfolio (P) GBFS configurations, and the PATTY solver. The top three scores are indicated by the cell colour intensity.

and a novelty heuristic n, we write $h_{\langle f,n\rangle}$ for the corresponding heuristic. The first entry in $f(\langle T\rangle, s)$ is given by the sequence of previously evaluated states during search. We let $3h$ denote the list of top 3 performing ENHSP heuristics h^{md}, h^{add}, and $h^{\mathrm{mrp}+\mathrm{hj}}$, $3n$ the list of their corresponding novelty heuristics $h^{\mathrm{md}}_{\langle\mathrm{B,QB}\rangle}$, $h^{\mathrm{add}}_{\langle\mathrm{B,QB}\rangle}$, and $h^{\mathrm{mrp}}_{\langle\mathrm{B,QB}\rangle+\mathrm{hj}}$ with $k = 2$, and $3h \| 3n$ the concatenation of these two lists. We choose the novelty heuristic extension $\langle\mathrm{B, QB}\rangle$ as it is the best forming novelty heuristic overall. Then $\mathrm{M}(\cdot)$ denotes a multi-queue search with a queue for each input heuristic, and $\mathrm{P}(\cdot)$ a portfolio with a uniform partitioning of time for each input heuristic. PATTY denotes the SMT planner by Cardellini, Giunchiglia, and Maratea (2024). All experiments are run with a 600 second timeout and 8GB memory limit on a single Intel Xeon 3.2 GHz CPU core. To help with the evaluation of the effectiveness of the outlined search techniques, we perform experiments to answer the following questions. We refer to the appendix for more detailed plots and tables.

How effective is $h^{\mathbf{md}}$? From Tab. 1, we notice that h^{md} is the best performing heuristic behind $h^{\mathrm{mrp}+\mathrm{hj}}$. This can be attributed to its fast evaluation speed rather than its informativeness as it generally expands more nodes than other heuristics on problems which both h^{md} and another heuristic solves. Exceptions to this rule are the fo-farmland and tpp domains in which delete relaxation heuristics perform poorly. Nevertheless it generally expands far fewer nodes than h^{gc}. Unfortunately, h^{md} returns the worst plan length on almost all problems compared to all other heuristics.

What is the best novelty heuristic for numeric planning? From Tab. 1, we notice that $h^{\mathrm{add}}_{\langle\mathrm{B,QB}\rangle}$ is the best performing novelty heuristic in terms of coverage. Observing the performance of configurations, the effect of choosing QB over PN for novelty heuristic is greater than choosing B over A for the novelty feature. This is supported by the fewer expansions of the QB variants compared to their PN counterparts over almost all domains. On the other hand, the comparison of plan length depends on the domain. The performance improvement over the base heuristic also depends on the domain. We lastly note that $k = 2$ almost always provides better performance than $k = 1$ for all novelty heuristics.

Is multi-queue search or using portfolios better for combining heuristics? Multi-queue search saves computation by reducing redundant node expansions and is able to make use of the exploration vs. exploitation paradigm when us-ing multiple diverse heuristics. On the other hand, portfolios make use of the fact that some heuristics perform better on some domains than others. From Tab. 2, we notice that portfolios overall achieve higher coverage even when using the simple uniform partitioning scheduling. The $\mathrm{P}(3h\|3n)$ configuration achieves strictly better coverage than $\mathrm{M}(3h\|3n)$ on 9 domains, and strictly lower coverage on 5 domains. This suggests that some domains can be solved efficiently by a single heuristic, while on other domains, we require the exploration effect of multi-queue search when none of the heuristics perform well. Furthermore, it depends on the domain whether the best heuristic or multi-queue search expands fewer nodes and returns better quality plans.

Which domains are more suited to constraint-based solvers and which to heuristic search? We note that the best performing constraint-based solver PATTY solves strictly more problems than $\mathrm{P}(3h\|3n)$ on 5 domains, and strictly fewer problems on 9 domains. The problems in which PATTY performs well generally have a high ratio of numeric variables and short rolled up plans, such as the fo-counters and fo-sailing domains. These problems have infinitely large state spaces and it is more likely for search-based methods to get lost. However, search-based methods perform better on domains which require long sequential plans or traversal over maps with many locations, such as delivery, drone, ext-plant-watering, markettrader and tpp. This is because SMT encodings may require many decision variables to represent problems with complex, sequential plans. However, we note that PATTY generally returns poorer quality plans with plan lengths up to an order of magnitude longer than search planners.

5 Conclusion

We have ported search techniques from classical planning to boost the performance of satisficing numeric planning solvers. We have formalised novelty heuristics for numeric planning, constructed a simple but powerful Manhattan distance heuristic, and extended the ENHSP planner with multi-queue search and naive portfolios. Experiments demonstrate the effectiveness of these techniques on the recent Numeric Track of the 2023 International Planning Competition (NT-IPC). Results also suggest a need for more diverse numeric planning benchmarks, as 78.8% of the problems from the NT-IPC are solved by our new $\mathrm{P}(3h\|3n)$ configuration in the ENHSP planner within 5 minutes, and 89.2% of the problems by either $\mathrm{P}(3h\|3n)$ or PATTY.

Acknowledgments

References

Arxer, J. E.; and Scala, E. 2023. International Planning Competition 2023 - Numeric Tracks. https://ipc2023-numeric.github.io. Accessed: 2024-02-19.

Bofill, M.; Espasa, J.; and Villaret, M. 2017. Relaxed Exists-Step Plans in Planning as SMT. In *IJCAI*, 563–570.

Cardellini, M.; Giunchiglia, E.; and Maratea, M. 2024. Symbolic Numeric Planning with Patterns. In *AAAI*, 20070–20077.

Coles, A.; Fox, M.; Long, D.; and Smith, A. 2008. A Hybrid Relaxed Planning Graph-LP Heuristic for Numeric Planning Domains. In *ICAPS*, 52–59.

Corrêa, A. B.; and Seipp, J. 2022. Best-First Width Search for Lifted Classical Planning. In *ICAPS*, 11–15.

Fawcett, C.; Helmert, M.; Hoos, H.; Karpas, E.; Röger, G.; and Seipp, J. 2011. FD-Autotune: Domain-Specific Configuration using Fast Downward. In *ICAPS 2011 Workshop on Planning and Learning*.

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

Garrett, C. R.; Kaelbling, L. P.; and Lozano-Pérez, T. 2016. Learning to Rank for Synthesizing Planning Heuristics. In *IJCAI*, 3089–3095.

Gigante, N.; and Scala, E. 2023. On the Compilability of Bounded Numeric Planning. In *IJCAI*, 5341–5349.

Helmert, M. 2002. Decidability and Undecidability Results for Planning with Numerical State Variables. In *AIPS*, 44–53.

Helmert, M. 2006. The Fast Downward Planning System. *J. Artif. Intell. Res.*, 191–246.

Helmert, M.; Röger, G.; and Karpas, E. 2011. Fast Downward Stone Soup: A Baseline for Building Planner Portfolios. In *ICAPS 2011 Workshop on Planning and Learning*.

Hoffmann, J. 2002. Extending FF to Numerical State Variables. In *ECAI*, 571–575.

Hoffmann, J. 2003. The Metric-FF Planning System: Translating "Ignoring Delete Lists" to Numeric State Variables. *J. Artif. Intell. Res.*, 24: 685–758.

Illanes, L.; and McIlraith, S. A. 2017. Numeric Planning via Abstraction and Policy Guided Search. In *IJCAI*, 4338–4345.

Katz, M.; Lipovetzky, N.; Moshkovich, D.; and Tuisov, A. 2017. Adapting Novelty to Classical Planning as Heuristic Search. In *ICAPS*, 172–180.

Katz, M.; Sohrabi, S.; Samulowitz, H.; and Sievers, S. 2018. Delfi: Online planner selection for cost-optimal planning.

Kuroiwa, R.; Shleyfman, A.; and Beck, J. C. 2023. Extracting and Exploiting Bounds of Numeric Variables for Optimal Linear Numeric Planning. In *ECAI*, 1332–1339.

Kuroiwa, R.; Shleyfman, A.; Piacentini, C.; Castro, M. P.; and Beck, J. C. 2021. LM-cut and Operator Counting Heuristics for Optimal Numeric Planning with Simple Conditions. In *ICAPS*, 210–218.

Kuroiwa, R.; Shleyfman, A.; Piacentini, C.; Castro, M. P.; and Beck, J. C. 2022. The LM-Cut Heuristic Family for Optimal Numeric Planning with Simple Conditions. *J. Artif. Intell. Res.*, 75: 1477–1548.

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

Lipovetzky, N.; and Geffner, H. 2012. Width and Serialization of Classical Planning Problems. In *ECAI*, 540–545.

Lipovetzky, N.; and Geffner, H. 2017. Best-First Width Search: Exploration and Exploitation in Classical Planning. In *AAAI*, 3590–3596.

Ma, T.; Ferber, P.; Huo, S.; Chen, J.; and Katz, M. 2020. Online Planner Selection with Graph Neural Networks and Adaptive Scheduling. In *AAAI*, 5077–5084.

Piacentini, C.; Castro, M. P.; Ciré, A. A.; and Beck, J. C. 2018. Linear and Integer Programming-Based Heuristics for Cost-Optimal Numeric Planning. In *AAAI*, 6254–6261.

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

Richter, S.; and Helmert, M. 2009. Preferred Operators and Deferred Evaluation in Satisficing Planning. In *ICAPS*, 273–280.

Richter, S.; and Westphal, M. 2010. The LAMA Planner: Guiding Cost-Based Anytime Planning with Landmarks. *J. Artif. Intell. Res.*, 39: 127–177.

Röger, G.; and Helmert, M. 2010. The More, the Merrier: Combining Heuristic Estimators for Satisficing Planning. In *ICAPS*, 246–249.

Scala, E.; Haslum, P.; Magazzeni, D.; and Thiébaux, S. 2017. Landmarks for Numeric Planning Problems. In *IJCAI*, 4384–4390.

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

Scala, E.; Haslum, P.; Thiébaux, S.; and Ramírez, M. 2016a. Interval-Based Relaxation for General Numeric Planning. In *ECAI*, 655–663.

Scala, E.; Haslum, P.; Thiébaux, S.; and Ramírez, M. 2020a. Subgoaling Techniques for Satisficing and Optimal Numeric Planning. *J. Artif. Intell. Res.*, 68: 691–752.

Scala, E.; Ramírez, M.; Haslum, P.; and Thiébaux, S. 2016b. Numeric Planning with Disjunctive Global Constraints via SMT. In *ICAPS*, 276–284.

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

Seipp, J.; Braun, M.; Garimort, J.; and Helmert, M. 2012. Learning Portfolios of Automatically Tuned Planners. In *ICAPS*, 368–372.

Shleyfman, A.; Kuroiwa, R.; and Beck, J. C. 2023. Symmetry Detection and Breaking in Linear Cost-Optimal Numeric Planning. In *ICAPS*, 393–401.

Teichteil-Königsbuch, F.; Ramírez, M.; and Lipovetzky, N. 2020. Boundary Extension Features for Width-Based Planning with Simulators on Continuous-State Domains. In *IJCAI*, 4183–4189.