Adapting Novelty to Classical Planning as Heuristic Search

Michael Katz
IBM Watson Health, Israel
katzm@il.ibm.com

Nir Lipovetzky
University of Melbourne, Australia
nir.lipovetzky@unimelb.edu.au

Dany Moshkovich
IBM Watson Health, Israel
mdany@il.ibm.com

Alexander Tuisov
Technion, Haifa, Israel
alexandt@campus.technion.ac.il

Abstract

The introduction of the concept of *state novelty* has advanced the state of the art in deterministic online planning in Atari-like problems and in planning with rewards in general, when rewards are defined on states. In classical planning, however, the success of novelty as the dichotomy between novel and non-novel states was somewhat limited. Until very recently, novelty-based methods were not able to successfully compete with state-of-the-art heuristic search based planners.

In this work we adapt the concept of novelty to heuristic search planning, defining the novelty of a state with respect to its heuristic estimate. We extend the dichotomy between novel and non-novel states and quantify the novelty degree of state facts. We then show a variety of heuristics based on the concept of novelty and exploit the recently introduced best-first width search for satisficing classical planning. Finally, we empirically show the resulting planners to significantly improve the state of the art in satisficing planning.

Introduction

Informative and fast heuristics as well as search-boosting and pruning techniques are crucial for the performance of heuristic search based planners. Recent years have seen considerable advancements in satisficing planning with the introduction of state-of-the-art heuristics (Keyder, Hoffmann, and Haslum 2014; Domshlak, Hoffmann, and Katz 2015), search boosting with multiple queues (Richter and Helmert 2009; Richter and Westphal 2010; Xie et al. 2014; Valenzano et al. 2014), and of states pruning techniques (Domshlak, Katz, and Shleyfman 2013; Lipovetzky and Geffner 2012). One such technique is based on the concept of *novelty* of a state, where the search procedure prunes nodes that do not qualify as *novel*. The concept has been successfully exploited in classical planning via SIW^+ and $DFS(i)$ search algorithms and in heuristic search, in conjunction with helpful actions (Lipovetzky and Geffner 2012; 2014; 2017). Novelty-based pruning was also successfully applied to blind state-space search for deterministic online planning in Atari-like problems (Lipovetzky, Ramirez, and Geffner 2015), where it was later generalized to account for rewards (Shleyfman, Tuisov, and Domshlak 2016; Jinnai and Fukunaga 2017). The latter work, although applied to Atari-like problems, is valid for planning with rewards in general, when rewards are defined on states.

In this work, we bring the concept of novelty back to heuristic search, which is guided by heuristic estimates of states rather than by rewards. We thus adapt the novelty definition of Shleyfman, Tuisov, and Domshlak (2016) to a novelty of a state with respect to its heuristic estimate. Specifically, we present the theoretical grounds for defining the novelty of states with respect to their heuristic estimates. Focusing on the individual facts, we start by introducing the dichotomic novelty notion per fact (rather than per state) and use it to quantify the novelty degree of states. This allows us not only to separate the *novel states* from the *non-novel* ones, which was the rationale behind the previous usage of novelty both in classical planning and in planning with rewards, but also to separate the novel states based on the degree of their novelty, and even to separate the non-novel states. Going even further, we quantify the contribution of the individual facts to the state novelty, extending the dichotomic notion. Our novelty notion is no longer used solely for pruning search nodes, but rather as a preference. We obtain new heuristic functions that are used for node ordering in a queue. Since these estimates are essentially goal-unaware, they can not be used as a single heuristic guidance to a best first search. Subsequently, we use novelty estimates to guide the search, breaking ties by classical goal-aware heuristic functions (Lipovetzky and Geffner 2017).

The rest of the paper is structured as follows. The background section presents the classical planning formalism and the definition of *reward novelty* (Shleyfman, Tuisov, and Domshlak 2016) which we build upon. The next section presents the theoretical grounds for defining the novelty of states with respect to their heuristic estimates. Next, we discuss the various possibilites of exploiting novelty estimates within the heuristic search framework. The experimental section presents an extensive empirical investigation of various novelty estimates and base heuristics, showing the benefits of exploiting novelty in practice. Next, we relate our work to a recently introduced notion of novelty of a state with respect to its heuristic estimate (Lipovetzky and Geffner 2017), discussing the similarities and the differences between the methods. Finally, we conclude and discuss possible future directions and prospects.

Background

We consider classical planning tasks $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_\star \rangle$ captured by the well-known SAS$^+$ formalism (Bäckström and Nebel 1995). In such a task, $\mathcal{V}$ is a set of finite-domain *state variables*, with each variable $v \in \mathcal{V}$ being associated with a finite domain $\mathcal{D}(v)$ of variable values. A *state* is a complete assignment to $\mathcal{V}$, and $\mathcal{S} = \prod_{v \in \mathcal{V}} \mathcal{D}(v)$ is the *state space* of Π. The complete assignment s_0 is the *initial state* of Π and the partial assignment $s_\star$ is the *goal* of Π. A state s is a *goal state*, denoted by $s \in \mathcal{S}_{s_\star}$, iff $s_\star \subseteq s$. $\mathcal{O}$ is a finite set of *operators*, each given by a pair $\langle pre, eff \rangle$ of partial assignments to $\mathcal{V}$, called *preconditions* and *effects*. Applying an operator o in a state s results in a state denoted by $s[\![o]\!]$. For a variable $v \in \mathcal{V}$ and a state s, the value of v in s is denoted by $s[v]$. For a variable $v \in \mathcal{V}$ and a value $\vartheta \in \mathcal{D}(v)$, the pair $\langle v, \vartheta \rangle$ is called a *fact*, and the set of all facts of Π is denoted by $\mathbb{F}$. We sometimes slightly abuse the notation and refer to a state s as to a set of facts, where $f \in s$ iff $f = \langle v, s[v] \rangle$ for some $v \in \mathcal{V}$.

As our work was inspired by the novelty in the scope of blind search with rewards for the Atari-like framework (Shleyfman, Tuisov, and Domshlak 2016), we present here the definition of novelty of a reward seen so far by Shleyfman, Tuisov, and Domshlak (2016). We reformulate the definition here in order to show how our definition relates to this one.

Definition 1 (Reward Novelty) *Given a reward function $R : \mathcal{S} \mapsto \mathbb{R}^{0+}$ and a set of states seen so far S, the* **novelty score** *of a fact (variable value) f is defined as*

$$N(f, S, R) = \begin{cases} \max\limits_{s \in S, f \in s} R(s), & f \in s \text{ for some } s \in S \\ -\infty, & \text{otherwise.} \end{cases}$$

Then, a state s is considered to be **novel** *iff $R(s) > N(f, S, R)$ for some $f \in s$.*

Although the definition is more general, covering sets of facts, for the sake of readability we restrict our attention here to individual facts.

Novelty Heuristics for Classical Planning

Our focus in this work is on classical planning as heuristic search. Thus, we start by adapting the aforementioned definition of novelty of a reward to heuristic search setting as follows.

Definition 2 (Heuristic Novelty) *Given a heuristic function $h : \mathcal{S} \mapsto \mathbb{R}^{0+}$ and a set of states seen so far S, the* **novelty score of a fact** *(variable value) f is defined as*

$$N(f, S, h) = \begin{cases} \min\limits_{s \in S, f \in s} h(s), & f \in s \text{ for some } s \in S \\ \infty, & \text{otherwise.} \end{cases}$$

Given a state s, the **novelty score of a fact** f **in state** s *is defined as $N(f, s, S, h) = N(f, S, h) - h(s)$ if $f \in s$.*

We say that a fact is *novel* in state s if its novelty score in s is strictly positive, and that a state is *novel* if it contains at least one novel fact. In what follows, we sometimes do not mention the heuristic h and the set of states seen so far S in the notation, where these are clear from the context, e. g., writing $N(f)$ instead of $N(f, S, h)$ and $N(f, s)$ instead of $N(f, s, S, h)$. Our goal here is to define heuristic functions based on a novelty score of facts in a state. Based on the definitions so far, we could define a variety of heuristics. Our first definition is rather basic, separating novel states from the rest. A heuristic $h_{\text{BN}}(s)$ is therefore defined as follows.

$$h_{\text{BN}}(s) = \begin{cases} 0, & \exists f \in s,\ N(f, s) > 0 \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

This heuristic corresponds to the definition of novelty by Shleyfman, Tuisov, and Domshlak (2016) and presents a basic dichotomy between states that are novel and those that are not. It is worth mentioning already here that even such a simplistic dichotomy leads to remarkable empirical improvements.

Observe that the domain of values h_{BN} can possibly obtain is extremely limited, h_{BN} stands for *binary novel* heuristic. All novel states are treated by the heuristic exactly the same. To alleviate the problem, we suggest to differentiate between novel states based on the number of novel facts in those states. Since heuristic functions are typically built to prefer lower values, the *quantified novel* heuristic h_{QN} is defined by

$$h_{\text{QN}}(s) = |\mathcal{V}| - \sum_{f \in s} N^{+}(f, s), \tag{2}$$

where $N^{+}(f, s)$ is 1 when $N(f, s) > 0$ and 0 otherwise. Similarly, let $N^{-}(f, s)$ be 1 when $N(f, s) < 0$ and 0 otherwise. Note that h_{QN}, as well as h_{BN} separate novel states from non-novel ones. The difference between the two heuristics is that h_{BN} does not separate novel states from each other, while h_{QN} does. Both h_{QN} and h_{BN} do not separate non-novel states though. Thus, h_{QN} dominates h_{BN} in its informativeness (assuming the same set of states seen so far S).

Our next heuristic is designed to separate non-novel states as well. For that, we account both for the number of possitive and negative values $N(f, s)$. The *quantified both* novel and non-novel heuristic h_{QB} is defined as

$$h_{\text{QB}}(s) = \begin{cases} h_{\text{QN}}(s), & h_{\text{QN}}(s) < |\mathcal{V}| \\ |\mathcal{V}| + \sum\limits_{f \in s} N^{-}(f, s), & \text{otherwise.} \end{cases} \tag{3}$$

Note that h_{QB} dominates h_{QN} in terms of informativeness, separating more states.

Quantifying Novelty Facts

The estimates we derived so far accounted only for whether the facts of the evaluated state are novel, assigning to each fact either 0 or 1. Similarly to how h_{QN} and h_{QB} extend the dichotomy between novel and non-novel states of h_{BN}, we would like to extend the dichotomy between novel and non-novel facts given by $N^{+}(f, s)$ and $N^{-}(f, s)$. Our aim here is to account for the value of $N(f, s)$, rather than merely for its sign. We present one variant of such extension, allowing

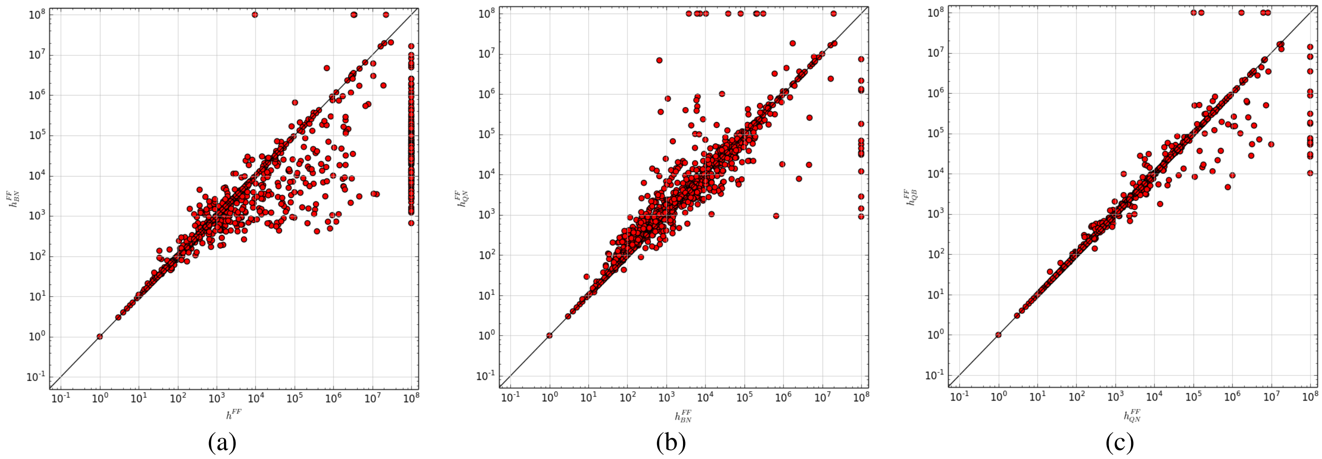


Figure 1: Evaluated nodes for (a) $h^{\text{FF}}_{\text{BN}}$ vs. h^{FF}, (b) $h^{\text{FF}}_{\text{QN}}$ vs. $h^{\text{FF}}_{\text{BN}}$, and (c) $h^{\text{FF}}_{\text{QB}}$ vs. $h^{\text{FF}}_{\text{QN}}$.

each fact to accept a range of values $K = \{0, \dots, k\}$ for some predefined constant k. We do that by mapping $N(f, s)$ to K via a simple function, setting

$$N^{+}_{k}(f,s) = \begin{cases} k & N(f,s) = \infty \\ \left\lceil \frac{k \times N(f,s)}{M_s} \right\rceil, & 0 < N(f,s) < \infty \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

and

$$N^{-}_{k}(f,s) = \begin{cases} -\left\lceil \frac{k \times N(f,s)}{M_s} \right\rceil, & N(f,s) < 0 \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where M_s is sufficiently large to warrant the image is within the range. The *quantified facts* heuristic h_{QF} is then defined as follows. Let $N^{op}_{k}(s) = \sum_{f \in s} N^{op}_{k}(f,s)$, where $op \in \{\text{+,-}\}$. Then

$$h_{\text{QF}}(s) = \begin{cases} k|\mathcal{V}| - N^{+}_{k}(s), & N^{+}_{k}(s) > 0 \\ k|\mathcal{V}| + N^{-}_{k}(s), & \text{otherwise.} \end{cases} \quad (6)$$

Having $M_s := \max(N(f), h(s))$ would result in a fact novelty score representing relative difference between the best-so-far heuristic value and the current one, while having M_s being the same for all states would represent the absolute difference. However, taking M_s to be too large would result in most facts obtaining values closer to 0 than to k. It is unclear how to obtain sufficently small, while still feasible constants. In our experiments, we chose to use the maximal base heuristic value observed so far (excluding dead ends). While not constant, this number rarely considerably increases during search and thus effectively allows us to obtain an absolute difference.

Multiple Base Estimators

Exploiting multiple estimators during search is a common practice in heuristic search planning, and many state of the art heuristic search planners exploit alternation between multiple queues (Richter and Westphal 2010; Katz and Hoffmann 2014). Each of the estimators can be enhanced by the novelty heuristic. However, different estimators can give different novelty scores to the same state, and what is the best way to integrate these scores is an open question. In what follows, we suggest a simple adaptation of the novelty heuristic to describe the novelty of a set of base estimators.

Definition 3 *Given a set of states seen so far S, a set of heuristic functions $H = \{h_1, \dots, h_n \mid h_i : \mathcal{S} \mapsto \mathbb{R}^{0+}\}$, and a state s, the novelty of a fact f in state s is defined as*

$$N(f, s, S, H) = \max_{h \in H} N(f, s, S, h) \quad \textit{if} \quad f \in s.$$

The heuristic functions h_{BN}, h_{QN}, h_{QB}, and h_{QF} defined in Eqs. 1–6 are obtained by replacing $N(f, s, S, h)$ with $N(f, s, S, H)$.

Exploiting Novelty Heuristic in Search

Note that our novelty heuristics are not goal-aware. Further, they all give a score of 0 to the initial state. Such heuristics are not meant to be used as a sole search guidance in a best first search, but rather together with an additional heuristic, that guide the search towards the goal. In what follows, we use the base heuristic function h as such guidance, especially since we have already paid the price of computing h.

There are several mechanisms that allow for co-exploiting several heuristic guidances in a single search. The two most well-known are *alternation* between multiple queues, henceforth denoted by $[x, y]$, and *tie breaking*. Naturally, for tie breaking, the order in which the ties are broken is extremely important. If the base heuristic is used first, breaking ties by a novelty heuristic, then the novelty heuristic will come into play only on the plateaus of the base heuristic, resulting in a similar behavior to a search with the base heuristic only.

In this work, our goal is different, we want to guide the search to examine novel states first. For that, we use the greedy best first search algorithm, using novelty heuristic as an initial heuristic, breaking ties by the base heuristic[1].

[1] Such scheme was also called best first width search (BFWS) (Lipovetzky and Geffner 2017).

This will essentially result in the same behaviour as if we defined several sequential queues to choose from. The heuristic $h_{\text{BN}}(s)$ basically emulates two queues, for *novel* states and for the rest, choosing from the non-novel queue only if the novel queue is empty. The heuristic h_{QN} emulates $|\mathcal{V}|+1$ queues, while $h_{\text{QB}}(s)$ emulates $2|\mathcal{V}|+1$ queues. Note that using h_{QF} allows to somewhat control the number of queues, varying the parameter k, getting up to $2k|\mathcal{V}|+1$ queues. Having too many queues, however, is a rather undesired behavior, since in the extreme case of all states having different heuristic values, the search is guided solely by the novelty heuristic, which as noted above, is not goal aware.

Experimental Evaluation

In order to evaluate the impact our novelty definition has on heuristic search planning, we implemented the three aforementioned heuristics within the Fast Downward planning framework (Helmert 2006). Our code is available upon request. All experiments are performed on an Intel(R) Core(TM) i7-3740QM CPU @ 2.70GHz, with a timeout of 30 minutes and a memory bound of 2GB, over the benchmark of STRIPS problems from the International Planning Competitions up to 2011.

First, we examine the effect of novelty on a single heuristic in a greedy best first search (GBFS) with lazy evaluation and no search enhancements such as helpful actions/preferred operators. We evaluate our four novelty variants h_{BN}, h_{QN}, h_{QB} , and h_{QF} on the commonly used heuristics FF (h^{FF}) (Hoffmann and Nebel 2001), landmarks count (h^{LM}) (Porteous, Sebastia, and Hoffmann 2001), and goal count (h^{GC}) (Fikes and Nilsson 1971). In what follows, we depict by h^X_Y the novelty heuristic where Y is the novelty variant and X is the base heuristic. We denote the tie breaking scheme by $h_1|h_2|\dots|h_n$. Further, since in what follows the novelty heuristic is always used as a first heuristic, breaking ties by other heuristics, h^X_Y also depicts the tie breaking scheme where the novelty heuristic is the first heuristic and the ties are broken by the base heuristic X. For example, $h^{\text{FF}}_{\text{BN}}$ describes the configuration where h_{BN} of h^{FF} is a first heuristic, tie breaking by h^{FF}, otherwise written by $h^{\text{FF}}_{\text{BN}}|h^{\text{FF}}$.

Table 1 depicts the per-domain coverage results for our first three variants of the novelty heuristic with h^{FF} and h^{LM}. Observe first that all novelty configurations perform much better in terms of the overall coverage than the base configurations h^{FF} and h^{LM}, with the maximal difference of 195 tasks for the best configuration $h^{\text{FF}}_{\text{QB}}$. Focusing first on the leftmost part of the table, first four columns that presents results for h^{FF} and the first three novelty variants with h^{FF} being the base heuristic, observe that there are multiple domains with a massive increase in coverage, such as PIPESWORLD-TANKAGE (18 tasks), ROVERS, TRANSPORT08, and VISIT-ALL (17 tasks each), BARMAN and PARKING (13 tasks each), PATHWAYS and PIPESWORLD-NOTANKAGE (12 tasks each), TRANSPORT11 (11 tasks), and TPP (10 tasks). Domains with a more "modest" increase include AIRPORT, PARC-PRINTER11, and TRUCKS (8 tasks each), as well as STORAGE (7 tasks), DEPOTS and NOMYSTERY (6 tasks each), MPRIME and PARC-PRINTER08 (4 tasks each). There are also domains where

Coverage		h^{FF}	$h^{\text{FF}}_{\text{BN}}$	$h^{\text{FF}}_{\text{QN}}$	$h^{\text{FF}}_{\text{QB}}$	$h^{\text{FF}}_{\text{QF}}$	h^{LM}	$h^{\text{LM}}_{\text{BN}}$	$h^{\text{LM}}_{\text{QN}}$	$h^{\text{LM}}_{\text{QB}}$
airport	50	31	39	38	39	**50**	29	32	32	37
barman11	20	4	14	14	17	9	**20**	13	13	13
depot	22	15	18	20	**21**	20	14	18	19	19
driverlog	20	18	**20**	**20**	**20**	**20**	18	18	18	19
elevators08	30	11	10	10	9	11	23	**30**	**30**	27
elevators11	20	0	0	0	0	0	5	**10**	**10**	**10**
floortile11	20	**8**	7	6	6	**8**	0	0	0	0
freecell	80	**80**	79	**80**	79	79	**80**	**80**	**80**	**80**
grid	5	4	**5**	**5**	**5**	4	4	4	4	**5**
logistics98	35	24	**25**	**25**	**25**	20	13	16	16	16
mprime	35	31	**35**	**35**	**35**	**35**	18	20	19	19
mystery	30	18	**19**	**19**	**19**	**19**	13	12	12	12
nomystery11	20	8	12	12	14	14	12	16	**17**	**17**
openstacks08	30	6	6	6	6	6	**30**	**30**	**30**	**30**
openstacks11	20	0	0	0	0	0	**20**	**20**	**20**	**20**
openstacks	30	**30**	**30**	**30**	**30**	24	**30**	**30**	**30**	**30**
parcprinter08	30	23	27	27	27	28	28	29	27	**30**
parcprinter11	20	7	15	15	15	16	18	**19**	**19**	**19**
parking11	20	7	**20**	**20**	**20**	6	0	3	5	10
pathways-nn	30	11	22	**23**	**23**	14	5	8	8	8
pegsol08	30	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	29
pegsol11	20	**20**	**20**	**20**	**20**	**20**	**20**	**20**	**20**	19
pipes-notank	50	30	41	41	**42**	**42**	37	36	36	33
pipes-tank	50	22	36	38	**40**	37	23	28	30	32
rovers	40	22	36	**39**	**39**	34	21	23	23	23
satellite	36	25	**26**	24	24	21	9	11	11	11
scanalyzer08	30	28	**30**	**30**	**30**	29	**30**	**30**	**30**	**30**
scanalyzer11	20	18	**20**	**20**	**20**	**20**	**20**	**20**	**20**	**20**
sokoban08	30	28	28	28	28	**29**	16	16	15	16
sokoban11	20	18	18	18	18	**19**	8	8	7	8
storage	30	19	**26**	25	**26**	**26**	17	20	22	22
tidybot11	20	16	17	17	17	15	**20**	**20**	**20**	19
tpp	30	20	**30**	**30**	**30**	25	23	28	27	27
transport08	30	11	28	28	28	16	**30**	**30**	**30**	**30**
transport11	20	0	10	11	11	1	18	**20**	**20**	**20**
trucks	30	14	17	18	**22**	18	7	8	7	7
visitall11	20	3	**20**	**20**	**20**	9	**20**	**20**	**20**	**20**
woodwork08	30	**30**	**30**	**30**	**30**	23	28	**30**	**30**	**30**
woodwork11	20	19	**20**	19	19	8	15	**20**	16	16
zenotravel	20	**20**	**20**	**20**	**20**	**20**	18	19	19	19
Sum	**1143**	729	906	911	**924**	825	790	845	842	852
Sum total	**1456**	1042	1219	1224	**1237**	1138	1103	1158	1155	1165

Table 1: Coverage for GBFS, comparing novelty variants h_{BN}, h_{QN}, h_{QB}, and h_{QF} for h, tie breaking on h to the base heuristic h for $h \in \{h^{\text{FF}}, h^{\text{LM}}\}$.

novelty confugurations seem to cause a slight coverage reduction, such as ELEVATORS08, FLOORTILE, FREECELL, and SATELLITE. Interestingly, in some of these domains, namely in ELEVATORS08 and SATELLITE novelty seems to contribute when used with h^{LM} as a base heuristic.

Focusing now on the rightmost part of the table, the h^{LM} based configurations, the overall increase in coverage when exploiting novelty is not as impressive as for h^{FF}, but is still very large. The overall coverage is increased by all novelty configurations, hitting the top of 62 tasks with $h^{\text{LM}}_{\text{QB}}$. There are still many domains with a large coverage increase, such as PARKING (10 tasks), PIPESWORLD-TANKAGE (9 tasks), AIRPORT (8 tasks), ELEVATORS08 (7 tasks), DEPOTS, ELEVATORS11, NOMYSTERY, STORAGE, TPP, and WOOD-

WORKING11 (5 tasks each). Further, the coverage reduction here can be quite large, e. g. 7 tasks in BARMAN domain and 4 tasks in PIPESWORLD-NOTANKAGE domain. The h^{FF} based novelty configurations excel in both of these domains.

In order to look beyond the per-domain coverage, Figure 1 pairwise compares the three novelty variants of h^{FF} from Table 1 and h^{FF} in terms of number of nodes evaluated during the search. Interestingly, the clearest win picture appears in Figure 1a, showing a comparison of h^{FF} to $h^{\mathrm{FF}}_{\mathrm{BN}}$, which is not the best performing variant in terms of overall coverage. Looking deeper into how the novelty heuristics improve one on top of another, Figure 1b shows that despite the better overall performance of $h^{\mathrm{FF}}_{\mathrm{QN}}$ compared to $h^{\mathrm{FF}}_{\mathrm{BN}}$, there is a sufficient amount of tasks where $h^{\mathrm{FF}}_{\mathrm{BN}}$ performs better, and in some cases even solves tasks where $h^{\mathrm{FF}}_{\mathrm{QN}}$ fails. The same is true when comparing $h^{\mathrm{FF}}_{\mathrm{QB}}$ to $h^{\mathrm{FF}}_{\mathrm{QN}}$ (Figure 1c), although here the win picture is more clear.

We have also experimented with the goal count heuristic. Here the picture is similar, with overall coverage increased by 59 tasks with $h^{\mathrm{GC}}_{\mathrm{BN}}$, by 73 tasks with $h^{\mathrm{GC}}_{\mathrm{QN}}$, and by 76 tasks with $h^{\mathrm{GC}}_{\mathrm{QB}}$, compared to the base configuration of greedy best first search with the goal count heuristic h^{GC}, which solved up to 913 tasks overall.

Quantifying Novelty Facts

In order to empirically test the benefits of extending novelty fact dichotomy, we implemented the h_{QF} heuristic in the same framework. For the reasons mentioned earlier, we use for M_s the maximal finite base heuristic value observed so far. In order to measure the effect of various k values, we experimented with $k \in \{1, 10, 100, 1000\}$. The best performance in terms of coverage was observed for $k = 100$. For the values of $k = 10$ and $k = 1000$, the coverage slightly decreases, with the worst performance observed for $k = 1$. For the lack of space, we report here the results for $k = 100$. The coverage results are depicted in the last column of the leftmost part of Table 1. While $h^{\mathrm{FF}}_{\mathrm{QF}}$ seems to perform overall worse than the first three novelty heuristic variants (1138 tasked solved overall, compared to 1219 for $h^{\mathrm{FF}}_{\mathrm{BN}}$), it outperforms h^{FF} overall by 96 tasks. $h^{\mathrm{FF}}_{\mathrm{QF}}$ outperforms other novelty variants on several domains, namely AIRPORT, ELEVATORS08, FLOORTILE, PARC-PRINTER08, PARC-PRINTER11, and SOKOBAN. It performs especially well in the AIRPORT domain, solving all 50 tasks. To our knowledge, no other domain-independent planner can solve all tasks in the AIRPORT domain.

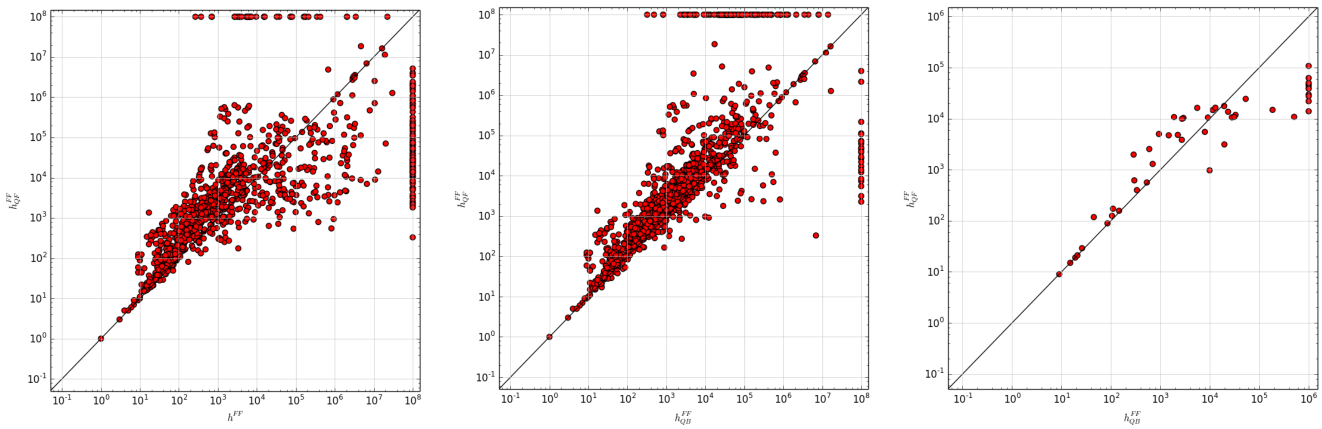
Figure 2 presents the results for the $h^{\mathrm{FF}}_{\mathrm{QF}}$ variant, comparing it to h^{FF} (Figure 2a) and to $h^{\mathrm{FF}}_{\mathrm{QB}}$ (Figure 2b) in terms of evaluated nodes. Looking beyond the coverage, Figure 2a hints that h^{FF} and $h^{\mathrm{FF}}_{\mathrm{QF}}$ have a complementary performance, showing that most tasks are positioned sufficiently far from the diagonal. Further, it shows a large amount of tasks that were solved by h^{FF} but not by $h^{\mathrm{FF}}_{\mathrm{QF}}$ and vice versa. Figure 2b, while indicating that $h^{\mathrm{FF}}_{\mathrm{QB}}$ performs better than $h^{\mathrm{FF}}_{\mathrm{QF}}$ overall, shows a considerable amount of tasks that were solved by the latter but not by the former. Finally, Figure 2c zooms in on the AIRPORT domain, on which $h^{\mathrm{FF}}_{\mathrm{QF}}$ demonstrates an exceptional performance.

Coverage	**FF|LM**	$*$|**FF|LM**			**LM|FF**	$*$|**LM|FF**		
		$h^{\mathrm{FF}}_{\mathrm{QB}}$	$h^{\mathrm{LM}}_{\mathrm{QB}}$	$h^{\mathrm{FF,LM}}_{\mathrm{QB}}$		$h^{\mathrm{LM}}_{\mathrm{QB}}$	$h^{\mathrm{FF}}_{\mathrm{QB}}$	$h^{\mathrm{FF,LM}}_{\mathrm{QB}}$
airport (50)	35	37	**42**	40	30	38	37	35
barman11 (20)	3	17	14	10	17	**20**	**20**	18
depot (22)	16	19	18	**21**	16	18	20	**21**
driverlog (20)	17	**20**	18	**20**	15	19	**20**	**20**
elevators08 (30)	10	13	29	29	13	29	12	**30**
elevators11 (20)	0	0	9	9	0	**10**	0	**10**
floortile11 (20)	4	**6**	4	5	3	4	5	3
freecell (80)	78	**80**	79	78	79	79	**80**	78
grid (5)	**5**	**5**	4	**5**	4	4	**5**	**5**
logistics98 (35)	22	26	20	**27**	19	20	**27**	**27**
mprime (35)	31	**35**	32	**35**	31	32	**35**	**35**
mystery (30)	18	18	18	**19**	17	18	18	18
nomystery11 (20)	8	14	**17**	16	10	**17**	15	16
openstacks11 (20)	**20**	16	**20**	**20**	**20**	**20**	16	**20**
parcprinter08 (30)	22	27	**30**	28	**30**	**30**	27	**30**
parcprinter11 (20)	7	15	**20**	16	**20**	**20**	15	**20**
parking11 (20)	13	**20**	**20**	**20**	**20**	**20**	**20**	**20**
pathways-nn (30)	9	23	17	**26**	19	17	24	**26**
pipes-notank (50)	33	**42**	33	**42**	34	31	**42**	**42**
pipes-tank (50)	24	39	31	40	26	33	39	**41**
rovers (40)	22	**38**	31	37	26	31	**38**	37
satellite (36)	21	25	13	**27**	18	13	25	26
scanalyzer08 (30)	28	**30**	**30**	**30**	27	28	**30**	28
scanalyzer11 (20)	18	**20**	**20**	**20**	17	18	**20**	18
sokoban08 (30)	28	28	**29**	28	**29**	**29**	28	28
sokoban11 (20)	18	18	**19**	18	**19**	**19**	18	18
storage (30)	18	**26**	21	25	17	21	25	25
tidybot11 (20)	15	17	18	17	14	**20**	18	17
tpp (30)	24	**30**	24	**30**	24	25	**30**	**30**
transport08 (30)	11	28	**30**	**30**	**30**	**30**	**30**	**30**
transport11 (20)	0	10	13	13	13	**20**	19	18
trucks (30)	15	**21**	18	18	15	17	**21**	18
visitall11 (20)	5	**20**	**20**	**20**	**20**	**20**	**20**	**20**
woodwork08 (30)	**30**	**30**	**30**	**30**	29	**30**	**30**	**30**
woodwork11 (20)	**18**	**18**	17	17	15	16	**18**	17
zenotravel (20)	17	**20**	**20**	**20**	18	**20**	**20**	**20**
Sum (1033)	663	851	828	886	754	836	867	**895**
Sum total (1456)	1086	1274	1251	1309	1177	1259	1290	**1318**

Table 2: Coverage for GBFS when either three novelty variants used, over h^{LM}, over h^{FF}, and over both, with ties broken first by h^{FF} and then by h^{LM} or viceversa.

Multiple Heuristics

Table 2 shows the effect of exploiting multiple base estimates within the novelty heuristic. Since these multiple estimates are computed for the benefit of novelty heuristic, we exploit them for tie breaking. The two base estimates we experimented with are h^{FF} and h^{LM}. Two base configurations in columns 1 and 5 are greedy best first search with h^{FF}, breaking ties by h^{LM}, denoted by FF|LM, and with h^{LM}, breaking ties by h^{FF}, denoted by LM|FF, respectively. All other configurations employ a novelty heuristic, breaking ties by either first h^{FF} and then h^{LM} (columns 2-4) or first h^{LM} and then h^{FF} (columns 6-8). Columns 2 and 7 employ $h^{\mathrm{FF}}_{\mathrm{QB}}$ as a first heuristic, while columns 3 and 6 start with $h^{\mathrm{LM}}_{\mathrm{QB}}$. Columns 4 and 8 describe the configurations with novelty heuristic based on both h^{FF} and h^{LM}, denoted by $h^{\mathrm{FF,LM}}_{\mathrm{QB}}$. In what follows, we refer to the configurations in the first 4 columns



(a) (b) (c)

Figure 2: Evaluated nodes for (a) $h^{\mathrm{FF}}_{\mathrm{QF}}$ vs. h^{FF}, (b) $h^{\mathrm{FF}}_{\mathrm{QF}}$ vs. $h^{\mathrm{FF}}_{\mathrm{QB}}$, and (c) $h^{\mathrm{FF}}_{\mathrm{QF}}$ vs. $h^{\mathrm{FF}}_{\mathrm{QB}}$ in Airport domain.

as the FF|LM scheme, and to those in the last 4 columns as the LM|FF scheme. When the scheme is clear from the context, since the tie breaking is the same across the configurations, we refer to the individual configurations by their novelty heuristic. For example, in the context of FF|LM scheme, the configuration in column 2 is referred to as $h^{\mathrm{FF}}_{\mathrm{QB}}$.

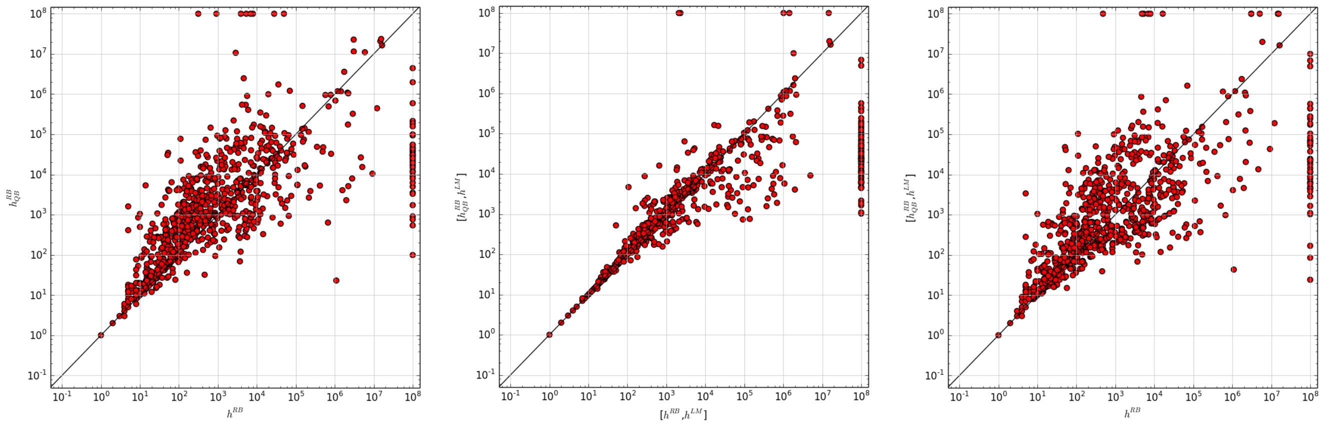
The overall coverage for the FF|LM scheme increases with more sophisticated novelty configurations. The best configuration, employing novelty of multiple heuristics increases coverage by 223 tasks compared to not using novelty at all and by 35 and 58 tasks compared to $h^{\mathrm{FF}}_{\mathrm{QB}}$ and $h^{\mathrm{LM}}_{\mathrm{QB}}$, respectively. For the LM|FF scheme, the picture is similar, with novelty of multiple heuristics increasing coverage by 141 tasks compared to not using novelty at all and by 28 and 59 tasks compared to $h^{\mathrm{FF}}_{\mathrm{QB}}$ and $h^{\mathrm{LM}}_{\mathrm{QB}}$, respectively. Looking at per domain coverage, in most domains taking both heuristics as a base for the novelty heuristic improves coverage compared to not using novelty. For the FF|LM scheme, the coverage is improved in 29 out of the 46 domains, sometimes dramatically (19 tasks each in Elevators08 and Transport08, 17 tasks in Pathways, 16 in Pipesworld-Tankage, 15 tasks each in Rovers and Visit-All, 13 tasks in Transport11, etc.), and is reduced by one instance in one domain. Similarly, for the LM|FF scheme, the coverage is improved in 26 domains and decreases in three domains by one instance each. Here also the increase in coverage is often dramatic, exemplified by 17 tasks in Elevators08 and 15 tasks in Pipesworld-Tankage.

Comparing to exploiting a single heuristic for novelty computation on a per domain basis, for the FF|LM scheme, the coverage increases in 15 domains and decreases in 7 compared to $h^{\mathrm{FF}}_{\mathrm{QB}}$ and increases in 14 domains and decreases in 9 compared to $h^{\mathrm{LM}}_{\mathrm{QB}}$. The largest decrease is 7 tasks in Barman compared to $h^{\mathrm{FF}}_{\mathrm{QB}}$ and 4 tasks each in Barman and Parc-Printer11 compared to $h^{\mathrm{LM}}_{\mathrm{QB}}$. The largest increase is by 16 tasks in Elevators08 compared to $h^{\mathrm{FF}}_{\mathrm{QB}}$ and 14 tasks in Satellite compared to $h^{\mathrm{LM}}_{\mathrm{QB}}$. In Satellite, however, this large increase is due to the poor performance of $h^{\mathrm{LM}}_{\mathrm{QB}}$ compared to the base configuration without novelty. For the LM|FF scheme, the coverage increases in 10 domains and decreases in 11 compared to $h^{\mathrm{FF}}_{\mathrm{QB}}$ and increases in 15 domains and decreases in 9 compared to $h^{\mathrm{LM}}_{\mathrm{QB}}$. The largest decrease is 3 tasks in both cases (Trucks for $h^{\mathrm{FF}}_{\mathrm{QB}}$ and Airport and Tidybot for $h^{\mathrm{LM}}_{\mathrm{QB}}$), while the largest increase is by 18 tasks in Elevators08 compared to $h^{\mathrm{FF}}_{\mathrm{QB}}$ and 13 tasks in Satellite compared to $h^{\mathrm{LM}}_{\mathrm{QB}}$, which is similar to the FF|LM scheme.

Novelty and State-of-the-art Heuristic Search

In order to test whether the novelty heuristic can contribute to the state of the art of heuristic search planning, we enhanced the *Mercury* planner (Katz and Hoffmann 2014) with our best performing variant, h_{QB} heuristic. Since we are interested in coverage, we focus on the first iteration of Mercury, which performs a greedy best first search with a red-black planning heuristic h^{RB} (Domshlak, Hoffmann, and Katz 2015), alternating with a queue ordered by preferred operators taken from the underlying h^{FF} heuristic. In our first variant, we apply h_{QB} to h^{RB}, breaking ties by h^{RB} (denoted by $h^{\mathrm{RB}}_{\mathrm{QB}}$). Here as well, the preferred operators were taken from the underlying h^{FF} heuristic. Further, we compare to a variant with additional queues ordered by the landmark count heuristic h^{LM} and preferred operators from h^{LM}, alternating between these queues, denoted by [h^{RB}, h^{LM}]. This variant is similar to the scheme employed by the *LAMA* planner, with the main difference being that h^{FF} is replaced by h^{RB}. Finally, we enhance this variant by using GBFS with $h^{\mathrm{RB}}_{\mathrm{QB}}$ instead of h^{RB} (denoted by [$h^{\mathrm{RB}}_{\mathrm{QB}}$, h^{LM}]). Applying novelty also to h^{LM} did not render better results in the last scheme. The coverage results are depicted in Table 3. Compared to *Mercury* (first column), the coverage is significantly increased by both configurations that employ the novelty heuristic (columns two and four). Comparing our best configuration (column four) to *Mercury*, the coverage decreases in two domains, Floortile and Rovers and increases on 17 domains, with overall increase in coverage by 43 tasks.

Figure 3 pairwise compares configurations from Table 3



(a) (b) (c)

Figure 3: Evaluated nodes for (a) $h^{\mathrm{RB}}_{\mathrm{QB}}$ vs. h^{RB}, (b) alternation between $h^{\mathrm{RB}}_{\mathrm{QB}}$ and h^{LM} vs. alternation between h^{RB} and h^{LM}, and (c) alternation between $h^{\mathrm{RB}}_{\mathrm{QB}}$ and h^{LM} vs. h^{RB}.

Coverage	h^{RB}	$h^{\mathrm{RB}}_{\mathrm{QB}}$	$[h^{\mathrm{RB}},h^{\mathrm{LM}}]$	$[h^{\mathrm{RB}}_{\mathrm{QB}},h^{\mathrm{LM}}]$
airport (50)	34	39	32	**43**
barman11 (20)	18	**20**	19	**20**
depot (22)	19	**20**	18	**20**
driverlog (20)	**20**	**20**	19	**20**
floortile11 (20)	**7**	**7**	4	3
freecell (80)	**80**	79	78	**80**
grid (5)	4	**5**	**5**	**5**
mprime (35)	**35**	**35**	31	**35**
mystery (30)	18	**19**	18	**19**
nomystery11 (20)	14	18	15	**19**
parking11 (20)	14	**20**	**20**	**20**
pathways-nn (30)	21	23	14	**26**
pipes-notank (50)	41	**43**	39	**43**
pipes-tank (50)	38	38	26	**41**
rovers (40)	**40**	36	32	37
scanalyzer08 (30)	28	**30**	**30**	29
scanalyzer11 (20)	18	**20**	**20**	19
sokoban08 (30)	26	**27**	26	**27**
sokoban11 (20)	16	**17**	16	**17**
storage (30)	20	**25**	17	**25**
tidybot11 (20)	15	17	15	**18**
tpp (30)	**30**	**30**	19	**30**
trucks (30)	15	**18**	15	**18**
woodwork11 (20)	**20**	19	**20**	**20**
Sum (722)	591	625	548	**634**
Sum total (1456)	1325	1359	1282	**1368**

Table 3: Coverage for GBFS with h^{RB}, $h^{\mathrm{RB}}_{\mathrm{QB}}$, alternation between h^{RB} and h^{LM} (marked by $[h^{\mathrm{RB}},h^{\mathrm{LM}}]$), and alternation between $h^{\mathrm{RB}}_{\mathrm{QB}}$ and h^{LM} (marked by $[h^{\mathrm{RB}}_{\mathrm{QB}},h^{\mathrm{LM}}]$).

in terms of evaluated nodes. Figure 3a compares the enhancement of h^{RB}, $h^{\mathrm{RB}}_{\mathrm{QB}}$ to h^{RB}. Our best performing configuration, alternation between $h^{\mathrm{RB}}_{\mathrm{QB}}$ and h^{LM} is compared to alternation between h^{RB} and h^{LM} (Figure 3b) and to h^{RB} (Figure 3c). Focusing on Figure 3c, depicting our best configuration vs. *Mercury*, there is no clear overall dominance of any of the configurations over the other, with the difference in the evaluated nodes getting up to four orders of magnitude in favor of novelty and up to three orders of magnitude in favor of *Mercury* in extreme cases. Note that most tasks are not on the diagonal, leaning towards one of the configurations, and thus the behaviour of these two configurations is complementary.

Recent Novelty-based Heuristic

Recently, a heuristic based on the notion of novelty was suggested (Lipovetzky and Geffner 2017). The core idea of the proposed heuristic is similar to ours. However, the definitions, although similar, vary a lot in the problem aspects tackled. In order to discuss the differences, we present here the definition of Lipovetzky and Geffner (2017). The novelty-based heuristic N_h is defined as

$$N_h(s) = \begin{cases} 0, & \exists f \in s:\ h(s) \notin \{h(s') \mid s' \in S, f \in s'\} \\ 1 & \text{otherwise.} \end{cases}$$

Observe that our binary novelty heuristic h_{BN} can also be viewed as

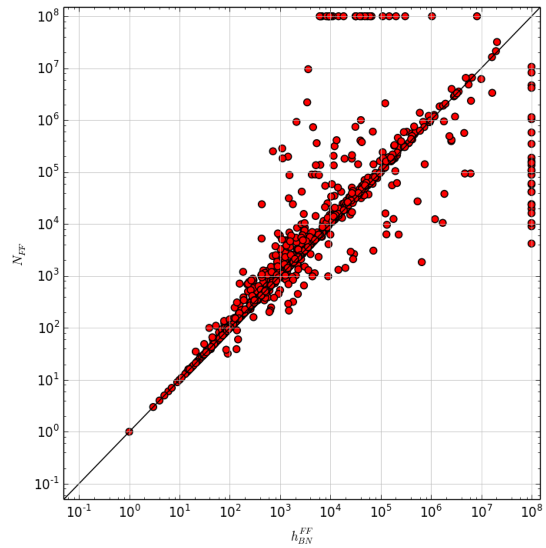
$$h_{\mathrm{BN}}(s) = \begin{cases} 0, & \exists f \in s:\ h(s) < \min\{h(s') \mid s' \in S, f \in s'\} \\ 1 & \text{otherwise.} \end{cases}$$

In words, the heuristic N_h defines a state to be novel if it achieves *some* heuristic value *for the first time* for at least one state fact. For instance, if heuristic values 3 and 5 have been achieved for a certain state fact, and now a value of 4 is observed, the state is considered to be novel. Thus, for each fact, all heuristic values observed so far are stored. For h_{BN} on the other hand, a state is novel if its heuristic value is the *best so far* for at least one state fact. As a result, only one value must be stored per fact.

It is worth mentioning that although the novelty-based heuristic suggested by Lipovetzky and Geffner does not go beyond the dichotomy between novel and non-novel states,

Coverage	N_{FF}	$h^{\text{FF}}_{\text{BN}}$
barman11 (20)	12	**14**
depot (22)	**22**	18
elevators08 (30)	**14**	10
elevators11 (20)	**1**	0
parcprinter08 (30)	26	**27**
parcprinter11 (20)	10	**15**
pipes-notank (50)	**46**	41
pipes-tank (50)	**42**	36
scanalyzer08 (30)	28	**30**
scanalyzer11 (20)	18	**20**
storage (30)	25	**26**
tidybot11 (20)	16	**17**
transport08 (30)	23	**28**
transport11 (20)	4	**10**
trucks (30)	**18**	17
Sum (422)	305	**309**
Sum total (1456)	1215	**1219**

(a)



(b)

Figure 4: Comparison between the basic novelty heuristic $h^{\text{FF}}_{\text{BN}}$ and the recently proposed heuristic N_{FF} in terms of (a) per-domain coverage, and (b) per-instance evaluated nodes.

it can be adapted to a quantitative measure of novelty, that separates novel states, analogously to the h_{QN} heuristic. It is unclear though whether an analogue of our h_{QB} heuristic that quantifies the non-novel states can also be derived.

Since N_h only separates novel states from non-novel ones, we compare it to our basic novelty heuristic h_{BN}, which also only separates novel states from non-novel ones. For that, we implemented the heuristic of Lipovetzky and Geffner in the same framework as our heuristics. The table in Figure 4a depicts the overall coverage comparison for GBFS with the two novelty variants, using h^{FF} heuristic as a base and for tie breaking. There are 15 domains in which coverage differs, of which 9 domains in favor of $h^{\text{FF}}_{\text{BN}}$, showing that the approaches are somewhat complementary. To demonstrate further that the approaches are complementary, Figure 4b shows a per instance comparison in terms of the number of evaluated nodes during search.

Conclusions and Future Work

We presented a quantitative notion of state novelty with respect to known heuristic estimates and suggested one way of exploiting this notion within heuristic search for satisficing classical planning. We differentiated not only how novel is a state but also how non-novel it can be. We further suggested that not all facts should contribute equally to the degree of state (non-)novelty. As a result, we showed how to derive multiple novelty heuristics integrated with other state-of-the-art goal aware heuristics, and tested their performance experimentally, finding them to perform extremely well. Finally, we demonstrated that novelty heuristics applied to the state-of-the-art heuristic search planner *Mercury* significantly improved its performance.

The presented concept of novelty opens up many interesting research directions. First, we suggested one way of quantifying the novelty of a fact, which is biased towards 0 values on larger base heuristic values. This can be an advantage on some domains, specifically we conjecture that it is the reason for the excellent performance in the AIRPORT domain, solving all 50 instances. But on many other domains it does not match the performance of our top novelty based performers. Thus, an investigation is needed on alternative quantification methods of state novelty in general, and the degree of contribution of each fact. Second, our current definition and implementation of handling multiple base estimates is rather straightforward. We explore one option of maximizing over the given heuristics. One could think of other aggregation methods. On the implementation side, we are required to store multiple estimates per fact, which can be quite memory inefficient. Other methods may be found to perform better. Third, the definitions of state novelty presented in this work operate with individual facts. Further, these definitions exploit the set of *all* individual facts. Both these restrictions are unnecessary. The definitions can be adapted to fact sets instead of individual facts, aggregating over arbitrary sets of such fact sets. This is a promising direction of great potential. There are many challenges here: how to derive informative and yet sufficiently small sets of fact sets; how to succinctly keep the "best estimates so far" for large sets of fact sets, and finally; how to aggregate the individual fact sets within an overall estimate. Last, but not least, extending the concept of heuristic novelty to richer formalisms is a promising direction for future research.

Acknowledgments. The work was performed, in part, while A. Tuisov was a summer intern at IBM Watson Health, Israel. The work by N. Lipovetzky is partially supported by the Australian Research Council linkage grant LP11010015.

References

Bäckström, C., and Nebel, B. 1995. Complexity results for SAS^+ planning. *Computational Intelligence* 11(4):625–655.

Domshlak, C.; Hoffmann, J.; and Katz, M. 2015. Red-black planning: A new systematic approach to partial delete relaxation. *Artificial Intelligence* 221:73–114.

Domshlak, C.; Katz, M.; and Shleyfman, A. 2013. Symmetry breaking: Satisficing planning and landmark heuristics. In Borrajo, D.; Kambhampati, S.; Oddi, A.; and Fratini, S., eds., *Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling (ICAPS 2013)*, 298–302. AAAI Press.

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

Helmert, M. 2006. The Fast Downward planning system. *Journal of Artificial Intelligence Research* 26:191–246.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 14:253–302.

Jinnai, Y., and Fukunaga, A. 2017. Learning to prune dominated action sequences in online black-box planning. In Singh, S., and Markovitch, S., eds., *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*.

Katz, M., and Hoffmann, J. 2014. Mercury planner: Pushing the limits of partial delete relaxation. In *Eighth International Planning Competition (IPC-8): planner abstracts*, 43–47.

Keyder, E.; Hoffmann, J.; and Haslum, P. 2014. Improving delete relaxation heuristics through explicitly represented conjunctions. *Journal of Artificial Intelligence Research* 50:487–533.

Lipovetzky, N., and Geffner, H. 2012. Width and serialization of classical planning problems. In De Raedt, L.; Bessiere, C.; Dubois, D.; Doherty, P.; Frasconi, P.; Heintz, F.; and Lucas, P., eds., *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, 540–545. IOS Press.

Lipovetzky, N., and Geffner, H. 2014. Width-based algorithms for classical planning: New results. In Schaub, T.; Friedrich, G.; and O'Sullivan, B., eds., *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, 1059–1060. IOS Press.

Lipovetzky, N., and Geffner, H. 2017. Best-first width search: Exploration and exploitation in classical planning. In Singh, S., and Markovitch, S., eds., *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*.

Lipovetzky, N.; Ramirez, M.; and Geffner, H. 2015. Classical planning with simulators: Results on the atari video games. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, 1610–1616.

Porteous, J.; Sebastia, L.; and Hoffmann, J. 2001. On the extraction, ordering, and usage of landmarks in planning. In Cesta, A., and Borrajo, D., eds., *Proceedings of the Sixth European Conference on Planning (ECP 2001)*, 174–182. AAAI Press.

Richter, S., and Helmert, M. 2009. Preferred operators and deferred evaluation in satisficing planning. In Gerevini, A.; Howe, A.; Cesta, A.; and Refanidis, I., eds., *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS 2009)*, 273–280. AAAI Press.

Richter, S., and Westphal, M. 2010. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research* 39:127–177.

Shleyfman, A.; Tuisov, A.; and Domshlak, C. 2016. Blind search for atari-like online planning revisited. In Kambhampati, S., ed., *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 3251–3257.

Valenzano, R.; Sturtevant, N. R.; Schaeffer, J.; and Xie, F. 2014. A comparison of knowledge-based GBFS enhancements and knowledge-free exploration. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling (ICAPS 2014)*, 375–379. AAAI Press.

Xie, F.; Müller, M.; Holte, R. C.; and Imai, T. 2014. Type-based exploration with multiple search queues for satisficing planning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI 2014)*. AAAI Press.