MERWIN Planner: Mercury Enchanced With Novelty Heuristic

Michael Katz  
IBM Research  
Yorktown Heights, NY, USA  
michael.katz1@ibm.com

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

Dany Moshkovich  
IBM Research  
Haifa, Israel  /mdany@il.ibm.com

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

Abstract
Heuristic search with red-black planning heuristics is among the most effective approaches to satisficing planning and the driving power behind the state-of-the-art satisficing planner Mercury. Another recent success in satisficing planning is due to the introduction of novelty based heuristic guidance, in particular a guidance measuring the novelty of a heuristic estimate in a state.

A satisficing planner that we baptize MERWIN empowers red-black planning heuristics with novelty based guidance, measuring the novelty of red-black planning heuristic estimates in explored states. MERWIN planner partitions the state space into novelty layers, expanding the most novel nodes first, and breaking ties within each layer by the red-black heuristic values.

Introduction
Delete relaxation heuristics are the key component of many successful planning systems (Bonet and Geffner 2001; Hoffmann and Nebel 2001; Richter and Westphal 2010). These heuristics though have a well-known pitfall of not being able to account for multiple achievements of the same fact, which leads to a wide research on how to take at least some deletes into account, e.g. (Fox and Long 2001; Gerevini, Saetti, and Serina 2003; Helmert 2004; Helmert and Geffner 2008; Baier and Botea 2009; Cai, Hoffmann, and Helmert 2009; Haslum 2012; Keyder, Hoffmann, and Haslum 2012). One such approach is so-called red-black planning (Domshlak, Hoffmann, and Katz 2015), where a subset of red state variables takes on the relaxed value-accumulating semantics, while the other black variables retain the regular semantics. This allows to interpolate between fully relaxed and regular planning. The work started with the introduction of the red-black framework and a theoretical investigation of tractability (Katz, Hoffmann, and Domshlak 2013a). Following up on this, practical non-admissible red-black plan heuristics were introduced, extending delete-relaxed plans into red-black plans (Katz, Hoffmann, and Domshlak 2013a). The technique, however, often suffered from dramatic overestimation incurred by following arbitrary decisions taken in delete-relaxed plans, and to overcome this shortcoming, Katz and Hoffmann (2013) refined the approach to rely less on such decisions, yielding a more flexible algorithm delivering better search guidance. Subsequently, Katz and Hoffmann (2013) refined the approach to rely less on such decisions, yielding a more flexible algorithm delivering better search guidance. Subsequently, Katz and Hoffmann (2014a) extended the approach to also handle conditional effects, enabling it to handle a range of planning problems. The latter work, although applied to Atari-like problems, is valid for planning with rewards in general, when rewards are defined on states. Consequently, Lipovetzky and Geffner (2017a) and Katz et al. (2017) brought the concept of novelty back to heuristic search, adapting the novelty definition of Shleyfman, Tuisov, and Domshlak (2016) to a novelty of a state with respect to its heuristic estimate. These two adaptations, although similar in nature, differ in detail (Katz et al. 2017). The new novelty notion was no longer used solely for pruning search nodes, but rather as a heuristic function, for node ordering in a queue. However, since such heuristics are not goal-aware, both Lipovetzky and Geffner (2014b) presented a red-black DAG heuristics for a tractable fragment characterized by DAG black causal graphs and devised some enhancements targeted at making the resulting red-black plans executable in the real task, stopping the search if they succeed in reaching the goal. Red-black DAG heuristics are in the heart of the Mercury planner (Katz and Hoffmann 2014a), the runner-up of the sequential satisficing track in the latest International Planning Competition (IPC 2014). It is worth mentioning that all aforementioned work on red-black planning handles the SAS+ fragment without conditional effects. Since conditional effects were a required feature to be supported by participating planners, Mercury handled conditional effects by simply compiling them away (Nebel 2000).

Search-boosting and pruning techniques have considerably advanced the state-of-the-art in planning as heuristic search (Richter and Helmert 2009; Richter and Westphal 2010; Xie et al. 2014; Valenzano et al. 2014; 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. Novelty has been successfully exploited for pruning in classical planning via $SIW^+$ and $DFS^i$ search algorithms (Lipovetzky and Geffner 2012; 2014). The blind novelty pruning $IV$ algorithm has shown great performance for classical online planning and finite horizon MDP problems over the Atari simulator (ALE) and General Video Game competition (GVG-AI) (Lipovetzky, Ramirez, and Geffner 2015; Bandres, Bonet, and Geffner 2018; Geffner and Geffner 2015), where it was later generalised to account for novelty based on 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. Consequently, Lipovetzky and Geffner (2017a) and Katz et al. (2017) brought the concept of novelty back to heuristic search, adapting the novelty definition of Shleyfman, Tuisov, and Domshlak (2016) to a novelty of a state with respect to its heuristic estimate. These two adaptations, although similar in nature, differ in detail (Katz et al. 2017). The new novelty notion was no longer used solely for pruning search nodes, but rather as a heuristic function, for node ordering in a queue. However, since such heuristics are not goal-aware, both Lipovetzky and Geffner (2014b) presented a red-black DAG heuristics for a tractable fragment characterized by DAG black causal graphs and devised some enhancements targeted at making the resulting red-black plans executable in the real task, stopping the search if they succeed in reaching the goal. Red-black DAG heuristics are in the heart of the Mercury planner (Katz and Hoffmann 2014a), the runner-up of the sequential satisficing track in the latest International Planning Competition (IPC 2014). It is worth mentioning that all aforementioned work on red-black planning handles the SAS+ fragment without conditional effects. Since conditional effects were a required feature to be supported by participating planners, Mercury handled conditional effects by simply compiling them away (Nebel 2000).
(2017a) and Katz et al. (2017) use the base heuristic as a secondary (tie-breaking) heuristic for node ordering. This general search framework is sometimes referred to as Best First Width Search (BFWS) (Lipovetzky and Geffner 2017a). Variants of BFWS can yield state-of-the-art polynomial planners (Lipovetzky and Geffner 2017b), and maintain good performance even when the action model is given as a Black-box simulator (Frances et al. 2017). In what follows, we exploit the notion of novelty of a state with respect to its heuristic estimate as defined by Katz et al. (2017).

In this work we construct a planner MERWIN, which stands for MERcury enhanced With Novelty, by exploiting both the red-black planning heuristic and the novelty of states with respect to a heuristic estimate. In particular, we modify the Mercury planner by replacing the queue ordered by red-black planning heuristic with a queue ordered by the novelty of a state with respect to its red-black planning heuristic estimate, breaking ties by that red-black planning heuristic estimate.

**Configurations**

MERWIN planner participates in three tracks, namely satisficing, agile, and bounded-cost. It is built on top of the Mercury planner (Katz and Hoffmann 2014a), runner-up of the sequential satisficing track of the International Planning Competition (IPC) 2014. Informally, as was mentioned above, the red-black planning heuristic in Mercury planner is enhanced by the novelty heuristic (Katz et al. 2017), and thus the queues ordered by the red-black planning heuristic in Mercury planner are replaced by the queues ordered by the novelty of a state with respect to its red-black planning heuristic estimate, with ties broken by the novelty heuristic estimate.

In what follows, we describe the heuristics used for all tracks and then detail the configuration for each track.

**Red-Black Planning Heuristic**

In order to describe the configuration of the red-black planning heuristic, we need to specify how a red-black task is constructed (which variables are chosen to be red and which black), also known as painting strategy, as well as how the red-black task is solved. In both cases, we followed the choices made by Mercury planner. Specifically, for red-black task construction followed one of the basic strategies, namely ordering the variables by causal graph level, and iteratively painting variables red until the black causal graph becomes a DAG (Domshlak, Hoffmann, and Katz 2015).

For solving the red-black task, MERWIN planner uses the algorithm presented in Figure 2 of Katz and Hoffmann (2014a). The algorithm receives a red-black planning task, as well as a set of red facts that is sufficient for reaching the red-black goals. Such a set is typically obtained from a relaxed solution to the task. Then, it iteratively (i) selects an action that can achieve some previously unachieved fact from that set, (ii) achieves its preconditions, and (iii) applies the action. Finally, when all the facts in the set are achieved, it achieves the goal of the task. We follow Katz and Hoffmann (2014a) in the two optimizations applied to enhance red-black plan applicability: selecting the next action in (i) giving a preference to actions whose black preconditions can be achieved without deleting facts from the set above, and selecting the sequences of actions in (ii), preferring those that are executable in the current state.

**Novelty Heuristic**

The novelty heuristic used in our planners measures the novelty of a state with respect to its red-black planning heuristic estimate, specificaly, we use the $h_{QB}$ heuristic, as described in Equation 3 of Katz et al. (2017). The quantified both novel and non-novel heuristic $h_{QB}$ is designed not only to distinguish novel states from non-novel ones, but also to separate the degree of (non-)novelty. Consequently, we use the best performing overall configuration of Katz et al. (2017) in MERWIN planner.

**Landmarks Count Heuristic**

Following the successful approaches of Mercury and LAMA planners, MERWIN planner uses additional queues ordered by the landmark count heuristic (Richter and Westphal 2010).

**Satisficing Track**

The configuration runs a sequence of search iterations of decreasing level of greediness. The first iteration is the greedy best-first search (GBFS) with deferred heuristic evaluation, alternating between four queues. The first queue is ordered by the novelty of a state with respect to its red-black planning heuristic estimate, with ties broken by $h_{RB}$. The second queue consists of states achieved by preferred operators of the red-black planning heuristic, ordered by $h_{RB}$, with ties broken by $h_{RB}$. The third and forth queues are ordered by the landmark count heuristic, with all successors and those achieved by the preferred operators, respectively.

The next iterations perform a weighted $A^*$ with deferred heuristic evaluation and decreasing weights $w = 5, 3, 2, 1$, continuing with $w = 1$. All these iterations alternate between the four queues as in Mercury planner, with the first two ordered by $h_{RB}$, with all successors and those achieved by the preferred operators, respectively, and the last two as in the first iteration. In case a solution is found in the previous iteration, its cost is passed as a pruning bound to the next iteration.

In case of non-unit costs, a cost transformation is performed, adding a constant 1 to all costs. Further, the first iteration is performed twice, once with unit costs and once with the increased costs.

**Agile Track**

The configuration in the agile track mimics the first iteration of the configuration in the satisficing track as described above.

These are basically the preferred operators of the full delete relaxation, an FF heuristic.


**Bounded-Cost Track**

The configuration in the bounded-cost track mimics the configuration in the agile track as described above. The only difference is that the cost bound is provided as an input.

**Supported Features**

As in the last competition, planners are required to support planning tasks with conditional effects. Following the strategy of *Mercury* planner, we have chosen here as well to compile the conditional effects away. This was done in a straightforward fashion, multiplying-out the actions (Nebel 2000) in the translation step. On one hand, this can lead to an exponential blow-up in the task representation size. On the other hand, it does not split up an operator application into a sequence of operator applications. Our decision was based on the success of the approach in *Mercury* planner, as well as the speculation that the latter option could potentially decrease red-black plan applicability, one of the main advantages of the current red-black heuristics.

**References**


