SYSU-Planner

Qiping Yang* and **Jiamin He*** and **Hankz Hankui Zhuo**
School of Data and Computer Science
Sun Yat-Sen University
Guangzhou, China
{yangqp5@mail2,hejm37@mail2,zhuohank@mail}.sysu.edu.cn

Abstract

In this paper, we describe the SYSU-Planner we have submitted to the SPARKLE Planning Challenge 2019. SYSU-Planner is a two-phased planner combining 1-BFWS and Forward-RHC. From best-first width search family, 1-BFWS is an incomplete but a really fast search algorithm pruning all the states with a novelty greater that one. For this reason, we use it as the front-end of our planner. However, in practice, it can hardly solve domains with numerous dead ends. Therefore we complement 1-BFWS with Forward-RHC. Forward-RHC is a modified search engine based on RHC (Refinement Hill Climbing), that would jump forward to the next state when learning stagnation occurs.

Introduction

In classical planning, searching utilizing heuristic is the dominant approach in the past 20 years (Bonet and Geffner 2001). In these methods based on heuristic search, delete relaxation is one of the most common and important strategies to compute heuristic, and it is equipped in a great portion of the winners of the International Planning Competition (IPC) in the past two decades (Hoffmann and Nebel 2001; Richter and Westphal 2010). However, some important characteristics of the planning problem are eliminated in the computation of heuristic based on delete relaxation making the planner hard to find the solution. In order to address this issue, diverse efforts on partial delete relaxation are made over the recent years (Fickert, Hoffmann, and Steinmetz 2016; Domshlak, Hoffmann, and Katz 2015).

One of the most influential efforts to prevent the important characteristics from deleting is using h^{CFF} heuristic (Fickert, Hoffmann, and Steinmetz 2016). The h^{CFF} heuristic maintains a list of conjunctions C , which were treated as atomic in the process of delete relaxation. Thus, some important characteristics are kept as conjunctions avoiding being departed by delete relaxation. As for the selection of the conjunctions C, (Fickert and Hoffmann 2017) have shown that it is more potent to generate and modify conjunctions during searches. They propose a family of online-refinement hill-climbing algorithms and show that with partial delete relaxation heuristic h^{CFF}, the algorithms are able to outperform state-of-the-art approaches in some IPC domains. It is also worth mentioning that an instance of the online refinement algorithm with h^{CFF} heuristic has taken part in classical tracks of IPC 2018 and is comparable to the winners in different tracks.

Recently, another family of heuristic search algorithms, width-based search algorithms, have arisen as a powerful planning method. The main difference between width-based search algorithms and other heuristic search algorithms is that width-based search algorithms utilize a notion of *novelty* (Geffner and Lipovetzky 2012). Different from other goal-oriented heuristic estimating the distance to the goal state, novelty is a kind of heuristic computing from the structural information of the current state. Heuristics based on novelty guide the search towards those states that are rather different from the states seen before and thus results in more exploration. Combining goal-oriented search and width-based search, best-first width search algorithms become a very effective approach towards planning (Lipovetzky and Geffner 2017a). They are strongly competitive in classical tracks of IPC 2018 (Frances et al. 2018), while one of the instances, BFWS-Preference planner wins the first price in the agile track.

The planner we submit to the SPARKLE Planning Challenge 2019, is much similar to the Dual-BFWS planner (Lipovetzky and Geffner 2017a). Same as Dual-BFWS, our planner is a two-phased planner with 1-BFWS as front-end. The difference between them is that we use a modified online refinement algorithm named Forward-RHC as the planner back-end, while Dual-BFWS uses an extension of BFWS(f_4) with $f_4 =< w, h_L, h_{ff} >$.

1-BFWS

The front-end of our planner runs 1-BFWS, which is a best-first width search algorithm making use of a notion of *novelty* (Lipovetzky and Geffner 2017b). Basically, the novelty of a state s is a measure of how novel the newly generated state s is compared to the states seen before. It is first introduced by (Geffner and Lipovetzky 2012). Later, (Lipovetzky and Geffner 2017a) extend the definition of novelty by incorporating goal-directed heuristics in the computation of novelty. With the extended notion of novelty, 1-BFWS prunes all the states with novelties greater than one when conducting the best-first search. This makes 1-BFWS an incomplete but fast search algorithm. Generally, 1-BFWS finds solutions in

*equal contribution

some tasks very fast or fails in others in a short period of time.

Forward-RHC

Nevertheless, some domains have a lot of deep dead ends, which makes search engines like BFWS perform poorly due to their exploration searching structure. This type of domains can be better solved by planners like OLCFF (Fickert and Hoffmann 2018), which employs an online learning heuristic h^{CFF} (Keyder, Hoffmann, and Haslum 2014). OLCFF can learn from different states by adjusting the conjunction set C of h^{CFF} when it encounters a local minimum. Refinement Hill Climbing (RHC), used as the search engine of OLCFF, will try to escape the local minima by adding conjunctions to C. However, OLCFF can potentially be stalled by local minima and spend most of the time trying to learn from the same state. Inspired by (Steinmetz and Hoffmann 2017), we try to address this problem by combining Depth First Search into the search engine called Forward-RHC. We allow the search engine to look forward to the next potential state, the neighboring state with the lowest h^{CFF}, when it encounters a learning stagnation, i.e. it fails to find a better state after certain iterations of learning processes. In doing so, our search engine can better refine its conjunction set C through learning from different states along the search path until it reaches a dead end, in which case the planner will restart from the initial state. An outline of the algorithm is listed in Algorithm 1.

Algorithm 1 Forward-RHC

```
procedure FORWARD-RHC
    s_p ← I
    s_best ← I
    count ← 0
    while G ∉ s do
        while count < limit do                          ▷ limit > 0
            S ← the set of states IW(C) visits from s_best
            s_p ← argmin_{s∈S} h(s)

            if h(s_p) < h(s_best) then
                s_best ← s_p
                count ← 0
                break
            else
                count ← count + 1
                refine h in s_best
            end if
        end while
        if count = limit then                           ▷ Look forward
            s_best ← s_p
        end if
        count ← 0
    end while
    return SOLVED
end procedure
```

Summary

We propose to combine two planners specialized in solving different domains: domains with few and simple actions and domains with many deep dead ends. In the resulting portfolio planner, we obtain the best of both worlds using the light and simple 1-BFWS first. As 1-BFWS would struggle in domains with a lot of deep dead ends, we apply the search engine Foward-RHC equipped with h^{CFF} on the same instance after the planner fails to reach the goal with 1-BFWS.

Acknowledgements. We thank the authors of 1-BFWS and OLCFF for allowing us to use the source code of the planners.

References

Bonet, B., and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence* 129(1-2):5–33.

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

Fickert, M., and Hoffmann, J. 2017. Complete local search: Boosting hill-climbing through online relaxation refinement. In *Twenty-Seventh International Conference on Automated Planning and Scheduling*.

Fickert, M., and Hoffmann, J. 2018. Olcff: Onlinelearning hcff. *IPC 2018 planner abstracts*.

Fickert, M.; Hoffmann, J.; and Steinmetz, M. 2016. Combining the delete relaxation with critical-path heuristics: A direct characterization. *Journal of Artificial Intelligence Research* 56:269–327.

Frances, G.; Geffner, H.; Lipovetzky, N.; and Ramirez, M. 2018. Best-first width search in the ipc 2018: Complete, simulated, and polynomial variants. *IPC2018–Classical Tracks* 22–26.

Geffner, H., and Lipovetzky, N. 2012. Width and serialization of classical planning problems.

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

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. 2017a. Best-first width search: Exploration and exploitation in classical planning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Lipovetzky, N., and Geffner, H. 2017b. A polynomial planning algorithm that beats lama and ff. In *Twenty-Seventh International Conference on Automated Planning and Scheduling*.

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

Steinmetz, M., and Hoffmann, J. 2017. State space search nogood learning: Online refinement of critical-path dead-end detectors in planning. *Artificial Intelligence* 245:1–37.