AIND Project #3
- Planning search -
FF: the Fast-Forward Planning System
(by Jörg Hoffmann)
research article review

Goals and techniques introduced

The article « FF: the Fast-Forward Planning System » by Jörg Hoffmann was published in the AI Magazine[1] in 2001. More details about the research work have been later made available on arXiv[2] : « The FF Planing System: Fast Plan Generation Through Heuristic Search » by Jörg Hoffmann and Bernhard Nebel (2001)

FF is quite similar to the Heuristic Search Planning (HSP) system introduced by B. Bonet and H. Geffner in 2000[3], and can be seen at one of HSP successors.

Like HSP, and as stated in its name, **FF relies on a forward search approach, and is guided by a heuristic function** to find the solution. Within this heuristic, the delete lists of actions are ignored in the relaxed problem.

However **FF differs from HSP on 3 points** :

1. Enhanced Heuristic Evaluation

FF is using a **relaxed version of Graphplan** and uses its output as the heuristic evaluation. Its evaluation are usually lower than HSP evaluations and is computed in Polynomial time (P)

for $i := 1, \ldots, m$ **do**
  $G_i := \{g \in \mathcal{G} \mid \text{layer-membership}(g) = i\}$
endfor
for $i := m, \ldots, 1$ **do**
  for all $g \in G_i$, g not marked TRUE at time i **do**
    select an action o with $g \in add(o)$ and layer membership $i-1$, o's difficulty being minimal
    for all $f \in pre(o)$, layer-membership$(f) \neq 0$, f not marked TRUE at time $i-1$ **do**
      $G_{\text{layer-membership}(f)} := G_{\text{layer-membership}(f)} \cup \{f\}$
    endfor
    for all $f \in add(o)$ **do**
      mark f as TRUE at times $i-1$ and i
    endfor
  endfor
endfor

Figure 1 : Relaxed Graphplan (From « The FF Planing System: Fast Plan Generation Through Heuristic Search » by Jörg Hoffmann and Bernhard Nebel (2001)

2. Novel Search Strategy

Like HSP, FF is using Hill Climbing search methods in order to minimise the number of nodes to evaluate, as this remains a costly operation even if it can be computed in Polynomial time. While HSP used a standard version of Hill Climbing algorithm, **FF is using an « *Enforced* Hill Climbing» version, where FF is evaluating all the direct successors** of a state. FF performs complete Breadth First Search, looking for successor with the better evaluation. This allows FF to better deal with the cases where Standard Hill Climbing algorithm fails: As stated in the paper[4] « Enforced Hill Climbing can often quite successfully solve tasks that do contain dead ends, as it does not necessarily get caught in one »

---

[1] https://www.aaai.org/ojs/index.php/aimagazine/article/view/1572/1471

[2] https://arxiv.org/abs/1106.0675 : « The FF Planing System: Fast Plan Generation Through Heuristic Search » by Hoffmann & Nebel (2001)

[3] Planning as Heuristic Search https://bonetblai.github.io/reports/ecp99-hspr.pdf

[4] https://arxiv.org/abs/1106.0675 : « The FF Planing System: Fast Plan Generation Through Heuristic Search » by Hoffmann & Nebel (2001)

initialize the current plan to the empty plan <>
$S := \mathcal{I}$
while $h(S) \neq 0$ **do**
    perform breadth first search for a state S' with $h(S') < h(S)$
    if no such state can be found **then**
      output "Fail", stop
    endif
    add the actions on the path to S' at the end of the current plan
    $S := S'$
endwhile

Figure 2 : Enforced Hill Climbing (From « The FF Planing System: Fast Plan Generation Through Heuristic Search » by Jörg Hoffmann and Bernhard Nebel (2001)

3. Helpful Actions

One other benefits of using the Relaxed Graphplan for the heuristics, is that in addition to provide a measure of distance to the goal, **the relaxed plan can also be used to identify successors**. This technique can then be **used to prune the search space.** (Note: in the full paper a second pruning technique is also mentioned : « Added goal deletion cuts out branches where some goal has apparently been achieved too early »).

This technique does not preserve completeness but proved to work well with the benchmark problems. Should the Enforced Hill-climbing method using the helpful actions pruning fails to find a solution, then the sytem falls back to a complete search algorithm like A*

Results

Fast Forward (FF) competed in the « Fully Automated Planner » category and was some of the best performing system in the International Planing Competition in 2000[5] : it won the **first prize with an « Exceptional Performance »** mention[6]

The authors have also investigated **which of the 3 major changes introduced in FF yields to better performance** in terms of runtime and path length. They have done this by comparing the performance on the 20 domains test suite of the system where each of the 3 changes can be activated/deactivated individually. Some of the conclusions are :

- FF distance evaluation improves runtime performance in half of domains, whatever the switch configuration
- Enforced Hill Climbing alone improved performance as many times as it degrades it, but combined with helpful actions it is faster in 16 of 20 domains.
- Enforced Hill Climbing often finds better solution than Hill Climbing
- Helpful actions improve the runtime performance in 3/4 of the domains, whatever the switch configuration

However, the authors also looked at the result with a bit of perspective, and criticised their solution, which remains simple by design. **They questioned the fact that the benchmarks used might just a also be too simple in structure**, and so they have explored the state spaces of the planning benchmarks, **leading to a taxonomy for planning domains[7]**, confirming by the way that most planning benchmarks belong to the « Simpler » category.

Such research, that helps to increase our understanding and bring new ideas to this field of AI, is important as it can turns into concrete benefits for some real world problems. In a world where AI is getting omni-present, from optimising supply chain to Bots assistants, but also in countless other areas, planning search proved to be an indispensable asset. Behind the high complexity of some problems, we have to compromise between solution optimality, completeness, runtime and computing requirements : any improvements in the fields can help to solve bigger problems, faster and better.

---

[5] See « Artificial Intelligence : A modern approach » book by S.J.Russel and Peter Norvig, chap. 10.4, Figure 10.11

[6] https://www.aaai.org/ojs/index.php/aimagazine/article/view/1571/1470 : Details and Results of AI'00 Planning competition in AI magazine

[7] https://pdfs.semanticscholar.org/1fea/a8ac047375bb58b51cfa3adf4d30a761052e.pdf : « Local Search Topology in Planning Benchmarks: An Empirical Analysis » (Hoffmann, 2001)