Empirical Evaluation of Search Algorithms for Satisficing Planning

Patrick von Reth

Department of Mathematics and Computer Science
Artificial Intelligence Group
University of Basel

2/9/2015
GBFS

- Best-first search:
  - $f(s)$ to find the most promising state to expand.

- GBFS:
  - $f(s) = h(s)$
Misleading heuristics

- Exploration of states not leading to a goal.
- Plateaus:
  - Many states are explored.
  - No improvement of $h(s)$.

Random Exploration:
- Explore random States from the open list.

Local Exploration:
- Start a search on a limited subset of states.
Search enhancements

Deferred evaluation:
- States are inserted with the heuristic value of their parent.
- Evaluated when they are explored.

Preferred Operators:
- Operators most probable part of a solution.
- Alternate open lists.
\( \epsilon \)-GBFS

- Extension of standard GBFS.
  - Probability \( \epsilon \) select a state uniformly randomly from the open list.
  - Probability \( 1 - \epsilon \) use standard behaviour of GBFS.
Type-based exploration

- States are inserted into buckets based on $h(s)$, $g(s)$, $const(1)$, ....
- Buckets are selected uniformly randomly as well as the states in the buckets.
- Used alternating with a standard open list.
Enforced hill climbing

- Standard GBFS until a better $h(s')$ value is found or the search fails.
- Run a new GBFS on state $s'$.
Monte-Carlo random walks

- Random exploration:
  - Multiple random walks:
    - Random operators are applied.
    - Only the end point is evaluated.
  - The path providing the best improvement is added to the global path.

- Configurations:
  - Helpful actions
  - Dead end avoidance
  - Iterative deepening
  - Acceptable progress
Local exploration

- Start a standard GBFS.
- If the heuristic value was not improved over a period of steps, start a local search.
- Depth of local search is limited.
- Close list is shared.
- Local search ends if:
  - the configured depth is reached.
  - a state \( s' \) with \( h(s') < h(s) \) is found.
  - the local search fails, the local open list is empty.
- Remaining states are merged.
- Alternate configuration: Local Random Walks
Diverse best-first search

- **Global open list:**
  - Probabilistic selection of states, based on their $h(s)$ and $g(s)$ value.
  - Smaller $g(s)$ and $h(s)$ are preferred.

- **Local open list:**
  - Standard open list.

- **Only local searches.**

- **Local search is limited by the initial $h(s)$.**

- **Remaining states are merged into the global open list.**

- **Next local search is started.**
All experiments were run on the same benchmark sets as in the original papers.

Results named base are those of a standard GBFS.
Results:

- Scale similar.

Two implementations:

- Bucket based
- Heap based
  - FIFO by ID.
### ε-GBFS

#### Action Table

<table>
<thead>
<tr>
<th>Action</th>
<th>RandomBucketOpenList</th>
<th>RandomOpenList</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert state</td>
<td>$O(1)$</td>
<td>$O(\log(n))$</td>
</tr>
<tr>
<td>Remove random state</td>
<td>$O(m)$</td>
<td>$O(\log(n))$</td>
</tr>
<tr>
<td>Remove min state</td>
<td>$O(1)$</td>
<td>$O(\log(n))$</td>
</tr>
</tbody>
</table>

### Graph

The graph compares the time usage of RandomBucketOpenList and RandomOpenList actions. The x-axis represents the time usage of RandomBucketOpenList in seconds, while the y-axis represents the time usage of RandomOpenList in seconds. The data points show a consistent trend where the time usage of RandomOpenList is generally higher than that of RandomBucketOpenList, with a notable range from $10^{-2}$ to $10^4$ seconds.
Results:
- Results scale similar.

Implementation:
- Reduced complexity $O(1)$ instead of $O(m)$ to the number of buckets.
  - Vector containing buckets.
  - Map pointing to buckets.
type-based-GBFS: Multiple heuristics

- ff-cea-g, ff-cg-cea-g, ff-cg-g are additions on our side.
- Longer keys lead to more evaluations resulting in worse results.
- Even the const(1) performs better.
Monte-Carlo random walks

- **Results:**
  - Number estimated from percentage results.
  - Good MHA results.

- **Implementation:**
  - Support for multiple configurations
    - Helpful actions
    - Dead end avoidance
    - Iterative deepening
    - Acceptable progress
Local exploration

- **Results:**
  - The results scale similar to the original results.

- **Implementation:**
  - Abstract wrapper
  - Combinations of different search engines possible.
### Results:
- Good results.
- Bad results for deferred evaluation.

### Implementation:
- Three open lists:
  - DiverseOpenList
  - ProbabilisticOpenList (global open list)
  - Any open list (local open list)
- ProbabilisticOpenList modified algorithm
  - Only iterate over existing values.
Comparison

- Comparison of all algorithms.
- On IPC 2011 benchmarks.
- Standard (eager) search.
- Deferred (lazy) search where applicable.
Eager

- All new algorithms improve results compared to standard GBFS.
- Random walks and EHC can not compete with the current algorithms.
- Simple randomisation leads to a similar improvement (\(\epsilon\)-GBFS, type-based-GBFS).
Deferred evaluation leads to worse results in most cases.
Conclusion

- All algorithms perform as good as announced.
- Simple randomisation can massively improve the results.
- For $\epsilon$-GBFS improvements showed their potential.
Future Work

- Try to combine.
- Try new configurations.
- We could try a single bucket randomisation with the alternating open list.
- Optimise.
- Comparison on a bigger benchmark set.