UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Simplified Memory-bounded A* Search

Alfons Juan
Jorge Civera

DSIC
Departament de Sistemes Informàtics i Computació

Objectives

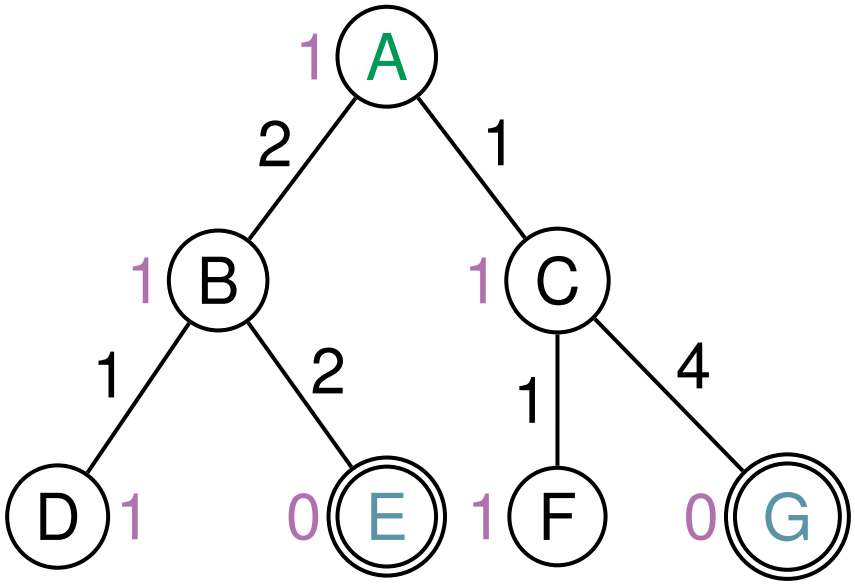
- To apply the SMA* algorithm.
- To build the SMA* search tree.
- To analyse properties, optimality and complexity of SMA* search.

Contents

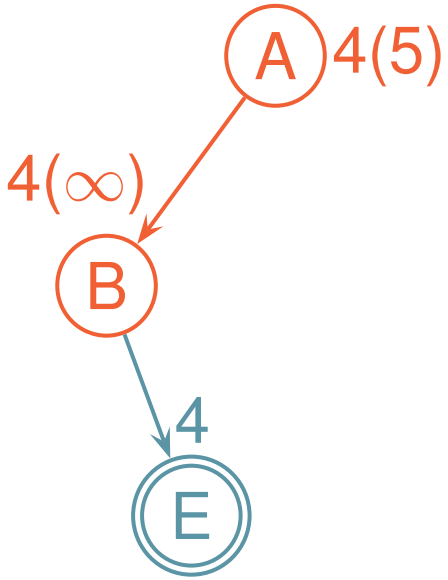
1 The SMA* by an example [1, 2]

SMA* behaves as A*, if enough memory available, otherwise delete unpromising node storing its ***f*** values and inserts new node.

State graph



SMA* (max. 3 nodes)



B's value is better than A's stored value, so it's the best, sol.!

2 Properties, optimality and complexity

- Control of repeated state to avoid loops, if enough memory
- ***Completeness:***
  - Yes, if enough memory to store the shallowest solution path
- ***Optimality:***
  - Yes, if enough memory to store the shallowest solution path
  - Otherwise, best solution with available memory
  - Search optimally efficient, if enough memory for full search tree
- ***Space complexity:*** User defined
- ***Temporal complexity:***
  - $O(b^d)$, in practice, extra cost to create and update nodes
- Good performance in explicit graphs with non-uniform costs

3 Conclusions

We have studied:

- The SMA* algorithm.
- The SMA* search space.
- Properties, optimality and complexity in SMA* search.

Some aspects to highlight on SMA*:

- Complete and optimum, if enough memory and h admissible.
- User-defined spatial cost.
- Temporal cost similar to A* with some extra cost in practice.

References

[1] Stuart J. Russell. Efficient memory-bounded search methods. In *Proc. of European Conference on Artificial Intelligence*, ECAI '92, page 1–5, USA, 1992. John Wiley & Sons, Inc.

[2] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentince Hall, first edition, 1995.