Agent-Centered Search: Situated Search with Small Look-Ahead

Sven Koenig

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
skoenig@cs.cmu.edu

Situated search is the process of achieving a goal in the world. Traditional single-agent search algorithms (such as the A* algorithm) usually assume completely known, stationary domains with deterministic actions. These assumptions favor search approaches that first determine open-loop plans (sequences of actions) that can then be executed blindly in the world. Consequently, single-agent search in AI is often performed in a mental model of the world (state space): states are represented as memory images and search is a process inside the computer (search-in-memory).

Situated search can interleave or overlap search-in-memory with action execution. This approach has advantages over traditional approaches in non-deterministic, non-stationary, or only partially known domains. It allows one, for example, to gather information by executing actions. This information can be used to resolve uncertainty caused by missing knowledge of the domain, inaccurate world models, or actions with non-deterministic effects. For instance, one does not need to plan for every contingency in non-deterministic domains; planning is necessary only for those situations that actually result from the execution of actions.

My research focuses on the design and analysis of agent-centered search methods. Agent-centered search methods are situated search methods that restrict the search-in-memory to a small part of the state space that is centered around the current state of the agent before committing to the execution of the next action. Since agent-centered search methods perform a small amount of search-in-memory around the current state of the agent, they can be characterized as "situated search with small look-ahead." Examples of agent-centered search methods include real-time heuristic search methods (they search forward from the current state of the agent), many on-line reinforcement learning algorithms, and exploration approaches.

A major difference between search-in-memory and agent-centered search is that search-in-memory methods can "jump around in the state space." If, after they have expanded a state, a state in a different part of the state space looks promising, they can expand this state next. In contrast, in situated search, if an agent wanted to explore a state, it would have to execute actions that get it there. The further the state is from the current state of the agent, the more expensive it is to get to that state. Furthermore, backtracking to a previous state might not be easy: it can be very expensive in state spaces with asymmetric costs and the agent might not even have learned how to "undo" an action execution. Thus, one can expect situated (agent-centered) search methods to have different properties than search-in-memory methods.

To date, agent-centered search methods have mostly been studied in the context of specific applications. The few existing results are mostly of empirical nature and no systematic or comparative studies have yet been conducted. It is therefore unclear when (and why) agent-centered search algorithms perform well. My research explores, both theoretically and experimentally, how agent-centered search methods behave. This includes an analysis of the factors that influence their performance. Such an analysis is helpful, for example, for predicting their performance in non-deterministic domains, for distinguishing easy from hard search problems, for representing situated search problems in a way that allows them to be solved efficiently, and for developing more adequate testbeds for agent-centered search methods than sliding tile puzzles, blocks worlds, and grid worlds. The second step then is to use these results to extend the functionality of agent-centered search algorithms to better fit typically encountered situated search situations. This includes probabilistic domains with non-linear reward structures, which can be caused, for instance, by the presence of dead-lines or risk attitudes. For more details and references to the literature, see (Koenig & Simmons 1996).

References

Koenig, S., and Simmons, R. 1996. Easy and hard testbeds for real-time search algorithms. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, this volume.