Using Priorities to Simplify Behavior Coordination

Brent E. Eskridge and **Dean F. Hougen**

Robotics, Evolution, Adaptation and Learning Laboratory (REAL Lab)

Artificial Intelligence Research Laboratories

School of Computer Science, University of Oklahoma

Previous research has used behavior hierarchies to address the problem of coordinating a large number of competing behaviors. However, behavior hierarchies have scaling problems since they require the state information of lower-level behaviors. The concept of abstracting this state information into **priorities** has recently been introduced to resolve this problem. In this work, we evaluate both the quality of priority-based behavior hierarchies and their ease of development. This is done by using grammatical evolution to learn how to coordinate low-level behaviors to accomplish a task. We show that not only do priority-based behavior hierarchies perform just as well as standard hierarchies, but that they promote faster learning of solutions that are better suited as components in larger hierarchies.

1. Motivation

- Behaviors allow real-time control of robots
- With a lot of behaviors, which one do you choose?
- A hierarchy of behaviors can help (Figure 1), *but*
  - Does **NOT** scale well to large numbers of behaviors
  - Interesting tasks have large numbers of behaviors

2. Approach

- Current work only abstracts output
- Input isn't abstracted at all (Figure 1)
- Why not abstract input as well?
- Abstract low-level input with a **PRIORITY**
- Composite behaviors use priority to weight sub-behaviors
- Can abstract low-level priorities into high-level priorities
- Creates a sensor hierarchy (Figures 2)

3. Experimental questions

- Do priorities provide too much abstraction?
- Does the abstraction hurt performance?
- Are behaviors that use priorities easier to learn?
- Can we use priorities to evaluate fitness?

4. Experimental setup

- Build simple composite behaviors with & without priorities
  - Collision avoiding, goal-seeking behavior
  - Collision avoiding, goal-seeking with run-away behavior
- Use grammatical evolution to evolve these behaviors
- Use three different fitness functions
  - Task-based
  - Priority-based
  - Combination of task & priority
- Train using training set of environments & each fitness function
- Evaluate using testing set of environments & task-based fitness function
- Compare with baseline naive heuristic & random behaviors

5. Results

- No statistically significant difference between priority & non-priority (Figures 3 & 7)
- Behaviors evolved using task & composite fitness functions outperform naive heuristic & random behaviors (Figures 3 & 7)
- Successful learning with priority-based fitness functions does not translate to success in the overall task (Figures 5 & 9)
- Priority-based behaviors appear to improve at a faster rate than non-priority ones (Figures 4, 6, 8 & 10)

6. Conclusions

- Using priorities does **NOT** hurt performance of composite behaviors
- Priorities **DO** aid in learning composite behaviors
- Priorities may allow scaling to more composite behaviors
- Task-based fitness function is required for success
- Future work
  - Use other machine learning techniques
  - Investigate priority-based fitness function failure
  - Test scalability with more complex behavior hierarchy

References

- Saffiotti, A., & Wasik, Z. (2003). Using hierarchical fuzzy behaviors in the robocup domain. In D. M. C. Zhou and D. Ruan (Eds.), Autonomous robotic systems, 235–262. Berlin, DE: Springer-Verlag.
- Tunstel, E. (2001). Fuzzy-behavior synthesis, coordination, and evolution in an adaptive behavior hierarchy. In A. Saffiotti and D. Driankov (Eds.), Fuzzy logic techniques for autonomous vehicle navigation, vol. 61 of Studies in Fuzziness and Soft Computing Series, Chapter 9. Physica-Verlag.

Acknowledgments

Thanks also go to all the members of the AI Research group for all their help. This material is based upon work supported by the U. S. Army Research Laboratory and the U. S. Army Research Office under grant number DAAD19-03-1-0142.