PHYS.ORG

# Streamlined rules for robots

June 13 2011, by Larry Hardesty

New techniques make it easier to calculate the optimal behaviors for fleets of autonomous robots, like these robot planes, working toward some common goal. Photo: NASA

With the explosion of the Internet and the commoditization of autonomous robots (such as the Roomba) and small sensors (such as the ones in most cell phones), computer scientists have become more and more interested in distributed computing, or how disparate autonomous devices — whether servers in a network or robots investigating an underwater oil spill — can work together toward some common goal.

Distributed devices have to adjust their behavior to changing circumstances. Frequently, however, their understanding of their circumstances is based only on a few local observations — and even those could be slightly inaccurate. Behavior that is perfectly reasonable in one case could prove catastrophic in another that, to a device, looks identical. Device programmers thus have to find behavioral policies that strike a balance between advancing the common goal and minimizing the risk of something going badly wrong.

Optimizing that balance would mean weighing every possible option for each device against all other options for all other devices under all circumstances. For even simple distributed-computing systems, that calculation quickly becomes so complex that it's basically insoluble. But Frans Oliehoek, a postdoc in MIT's Computer Science and Artificial Intelligence Laboratory, is developing new techniques to calculate policies for distributed-computing systems. Although those techniques aren't guaranteed to find the perfect policy, they will usually come pretty close — and they won't take centuries to yield an answer.

To get a clearer idea of the problem, consider a very simple example. Companies such as Google or Facebook maintain server farms with tens of thousands of computers. There's a lot of redundant information on those computers, so that hordes of users can access the same information at the same time. If a given computer is falling behind in handling users' requests, how long should it let its queue of unanswered requests get before it fobs them off on another computer? Ten? Fifteen? A thousand? A million? The optimal answer has to strike a balance among cases where the other servers in the farm are idle, cases where the other servers have even longer queues, and everything in between. A given server may be able to infer something about traffic as a whole by glancing at the queues of the servers next to it. But if it were continually asking all the other servers in the farm about the length of their queues, it would choke the network with queries.

## Historical perspective

Making the problem even more complicated, policy has to vary according to a device's history. It may be, for instance, that a robot helicopter trying to find a way into a burning building is much less likely to get itself incinerated if it makes two reconnaissance loops around the building before picking an entry point than if it makes just one. So its policy isn't as simple as, "If you've just completed a loop, fly through the window farthest from the flames." Sometimes it's, "If you've just completed a loop, make another loop." Moreover, if a squadron of helicopters is performing a collective task, the policy for any one of them has to account for all the possible histories of all the others.

In a series of papers presented at the International Conference on Autonomous Agents and Multiagent Systems, Oliehoek and colleagues at several other universities have described a variety of ways to reduce the scale of the policy-calculation problem. "What you want to do is try and decompose the whole big problem into a set of smaller problems that are connected," Oliehoek says. "We now have some methods that seem to work quite well in practice."

The key is to identify cases in which structural features of the problem mean that certain combinations of policies don't need to be evaluated separately. Suppose, for instance, that the goal is to find policies to prevent autonomous helicopters from colliding with each other while investigating a fire. It could be that after certain sequences of events, there's some possibility of helicopter A hitting helicopter B, and of helicopter B hitting helicopter C, but no chance of helicopter A hitting helicopter C. So preventing A from colliding with C doesn't have to factor in to the calculation of the optimal policy. In other cases, it's possible to lump histories together: Different histories can still point to the same result for the same action.

The mathematical model of decision making that Oliehoek has been investigating "is a very general model, so you can model all sorts of decision problems with it," says Francisco Melo, an assistant professor of computer science and engineering at Portugal's Universidade Técnica de Lisboa. “It’s a very lively line of work right now.” But, Melo, adds, “it’s a very complex model. There’s not much hope of computing an exact solution except for very, very, very small problems.” Melo says that while other researchers have performed theoretical analyses of the complexity of the model, and still others have attempted to find practical algorithms that yield approximations of the ideal policy, Oliehoek’s work combines the virtues of both lines of research. “I think that Frans’ work is all trying to — from a theoretical point of view — understand, if we actually want to do planning, what sorts of structures can we explore?” Melo says. “And that is also useful when you’re trying to make approximate algorithms. So I think that his contributions were important.”

*This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.*

Provided by Massachusetts Institute of Technology

Citation: Streamlined rules for robots (2011, June 13) retrieved 11 May 2024 from https://phys.org/news/2011-06-robots.html