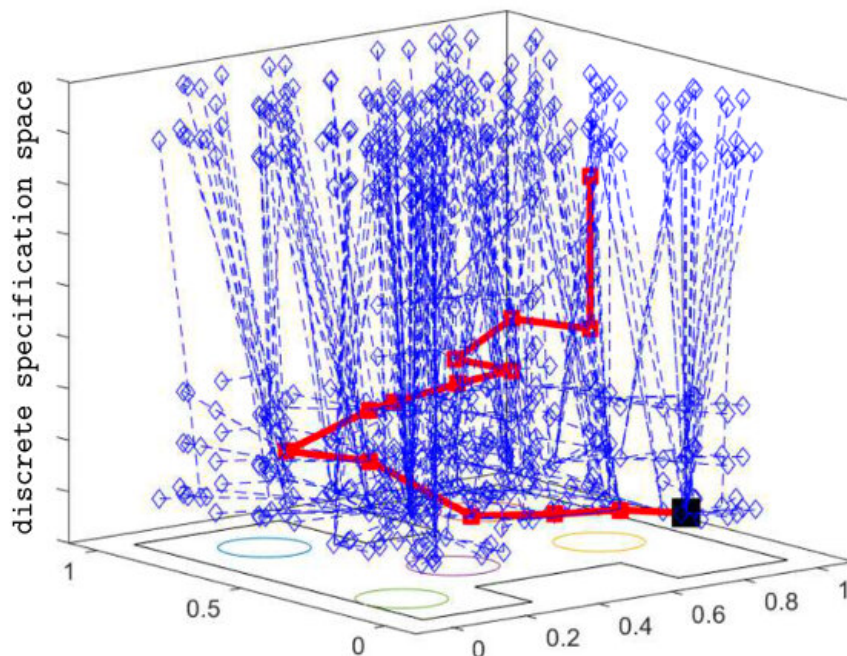


# Coordinating complex behaviors between hundreds of robots

July 1 2020, by Ken Kingery



As a control algorithm moves robots through a space, each state the robots can be in is represented by a spot on the 3D graph. As the algorithm explores new options, it finds failures and potential next moves. This graph represents all of the wrong moves explored (blue) and the one route that is correct (red). Credit: Michael Zavlanos, Duke

In one of the more memorable scenes from the 2002 blockbuster film *Minority Report*, Tom Cruise is forced to hide from a swarm of spider-

like robots scouring a towering apartment complex. While most viewers are likely transfixed by the small, agile bloodhound replacements, a computer engineer might marvel instead at their elegant control system.

In a building several stories tall with numerous rooms, hundreds of obstacles and thousands of places to inspect, the several dozen robots move as one cohesive unit. They spread out in a search pattern to thoroughly check the entire building while simultaneously splitting tasks so as to not waste time doubling back on their own paths or re-checking places other robots have already visited.

Such cohesion would be difficult for human controllers to achieve, let alone for an artificial controller to compute in [real-time](#).

"If a control problem has three or four robots that live in a world with only a handful of rooms, and if the collaborative task is specified by simple logic rules, there are state-of-the-art tools that can compute an optimal solution that satisfies the task in a reasonable amount of time," said Michael M. Zavlanos, the Mary Milus Yoh and Harold L. Yoh, Jr. Associate Professor of Mechanical Engineering and Materials Science at Duke University.

"And if you don't care about the best solution possible, you can solve for a few more rooms and more complex tasks in a matter of minutes, but still only a dozen robots tops," Zavlanos said. "Any more than that, and current algorithms are unable to overcome the sheer volume of possibilities in finding a solution."

In a new paper published online on April 29 in the *International Journal of Robotics Research*, Zavlanos and his recent Ph.D. graduate student, Yiannis Kantaros, who is now a postdoctoral researcher at the University of Pennsylvania, propose a new approach to this challenge called STyLuS\*, for large-Scale optimal Temporal Logic Synthesis, that can

solve problems massively larger than what current algorithms can handle, with hundreds of robots, tens of thousands of rooms and highly complex tasks, in a small fraction of the time.

To understand the basis of the new approach, one must first understand linear temporal logic, which is not nearly as scary as it sounds. Suppose you wanted to program a handful of robots to collect mail from a neighborhood and deliver it to the post office every day. Linear temporal logic is a way of writing down the commands needed to complete this task.

For example, these commands might include to visit each house in sequential order, return back to the post office and then wait for someone to retrieve the collected mail before setting out again. While this might be easy to say in English, it's more difficult to express mathematically. Linear temporal logic can do so by using its own symbols which, although might look like Klingon to the common observer, they're extremely useful for expressing complex control problems.

"The term linear is used because points in time have a unique successor based on discrete linear model of time, and temporal refers to the use of operators such as until, next, eventually and always," said Kantaros.

"Using this mathematical formalism, we can build complex commands such as 'visit all the houses except house two,' 'visit houses three and four in sequential order,' and 'wait until you've been to house one before going to house five.' "

To find [robot](#) controllers that satisfy such complex tasks, the location of each robot is mapped into a discrete data point called a "node." Then, from each node, there exist multiple other nodes that are a potential next step for the robot.

A traditional controller searches through each one of these nodes and the potential paths to take between them before figuring out the best way to navigate its way through. But as the number of robots and locations to visit increase, and as the logic rules to follow become more sophisticated, the solution space becomes incredibly large in a very short amount of time.

A simple problem with five robots living in a world with ten houses could contain millions of nodes, capturing all possible robot locations and behaviors towards achieving the task. This requires a lot of memory to store and processing power to search through.

To skirt around this issue, the researchers propose a new method that, rather than constructing these incredibly large graphs in their entirety, instead creates smaller approximations with a tree structure. At every step of the process, the algorithm randomly selects one node from the large graph, adds it to the tree, and rewires the existing paths between the nodes in the tree to find more direct paths from start to finish.

"This means that as the algorithm progresses, this tree that we incrementally grow gets closer and closer to the actual graph, which we never actually construct," said Kantaros. "Since our incremental graph is much smaller, it is easy to store in memory. Moreover, since this graph is a tree, graph search, which otherwise has exponential complexity, becomes very easy because now we only need to trace the sequence of parent nodes back to the root to find the desired path."

It had been long accepted that growing trees could not be used to search the space of possible solutions for these types of robot control problems. But in the paper, Zavlanos and Kantaros show that they can make it work by implementing two clever tricks. First, the algorithm chooses the next node to add based on information about the [task](#) at hand, which allows the tree to quickly approximate a good solution to the problem. Second,

even though the algorithm grows trees, it can still detect cycles in the original graph space that capture solutions to such temporal logic tasks.

The researchers show that this method will always find an answer if there is one, and it will always eventually find the best one possible. They also show that this method can arrive at that answer exponentially fast. Working with a problem of 10 robots searching through a 50-by-50 grid space— 250 houses to pick up mail—current state-of-the-art algorithms take 30 minutes to find an optimal solution.

STyLuS\* does it in about 20 seconds.

"We have even solved problems with 200 robots that live on a 100-by-100 grid world, which is far too large for today's algorithms to handle," said Zavlanos. "While there currently aren't any systems that use 200 robots to do something like deliver packages, there might be in the future. And they would need a control framework like STyLuS\* to be able to deliver them while satisfying complex logic-based rules."

**More information:** Yiannis Kantaros et al, STyLuS\*: A Temporal Logic Optimal Control Synthesis Algorithm for Large-Scale Multi-Robot Systems, *The International Journal of Robotics Research* (2020). [DOI: 10.1177/0278364920913922](https://doi.org/10.1177/0278364920913922)

Provided by Duke University

Citation: Coordinating complex behaviors between hundreds of robots (2020, July 1) retrieved 30 May 2023 from <https://techxplore.com/news/2020-07-complex-behaviors-hundreds-robots.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private

study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.