Smart Swarms Seek New Ways to Cooperate

New algorithms show how swarms of very simple robots can be made to work together as a group.

*By Kevin Hartnett*

In a lab at the Georgia Institute of Technology, physicists run experiments with robots that look as though they came from the dollar store. The robots can’t move through space. They can’t communicate. Mostly they flap their little arms, like beetles stuck on their backs.

But put a lot of these objects together and you get something from nothing: They hit each other, nudge each other and tangle with each other. And eventually, they start to work as a unit.

Researchers are learning how to control these systems so that they function in a manner similar to swarms of bees or colonies of ants: Each individual operates in response to the same basic set of instructions. But when the swarm comes together, its members can carry out complex behaviors without any centralized direction.

“Our whole perspective is: What’s the simplest computational model that will achieve these complicated tasks?” said Dana Randall, a computer scientist at Georgia Tech and one of the lead researchers on the project. “We’re looking for elegance and simplicity.”

As a computer scientist, Randall thinks about the problem in algorithmic terms: What is the most basic set of instructions individual elements in a swarm can run, based on the meager data they can collect, that will lead inevitably to the complex collective behavior researchers want? This past November, Randall and colleagues published an algorithm that ensures that an idealized particle swarm will move in a coordinated manner.

The work with these robots, known as “smarticles,” is part of a broader interest in the feasibility and applications of self-organizing robots. Other examples include “droplet”-size robots being developed at the University of Colorado, “Kilobot” swarms at Harvard University, and swarmanoids out of a pioneering lab in Belgium. In many of these cases the idea is to mimic emergent phenomena found in nature, like the regimented motion of a decentralized colony of army ants or the unconscious, self-programming assembly of DNA molecules.

“We know what we want the collective to do, but in order to program it we need to know what each agent must be doing on the individual level,” said Melvin Gauci, a researcher at Harvard working on swarm robotics. “Going between those two levels is what’s very challenging.”
**Beware of Leaders**

Daniel Goldman is a physicist at Georgia Tech who is leading the experiments with smarticles (a portmanteau of “smart active particles”). His fundamental scientific interest is in the physics of active granular materials that have the ability to change their own shape. In a slide deck he brings to conferences, he includes a clip from *Spider-Man 3* that shows **the birth of the supervillain Sandman**: Loose grains of sand skitter across the desert and then congeal into the shape of a man. Smarticles are Goldman’s way of testing active granular materials in a lab.

“They give us a way to use geometry to control the properties of a material. If you blur your eyes, you can think of this collection of smarticles as a real material,” Goldman said.

The smarticles have short arms, like the prongs of a staple, that they can swing back and forth. They respond to light and tones of different frequencies. They can also be programmed to adjust the rate at which they swing their arms in response to the other smarticles they encounter in their immediate vicinity.

There are a few basic maneuvers you might like a swarm of smarticles to perform: compression (packing together), expansion (spreading out) and locomotion. These maneuvers could serve as building blocks for more complicated feats, but even the most basic functions, like compression, are hard to engineer when none of the smarticles have any idea where they’re positioned in relation to the overall group.

To understand the opportunity and the challenge of engineering complex behavior from simple parts, it’s worth keeping in mind what an individual smarticle knows. And the answer is: not much. It can’t see, it has limited memory, and the only thing it knows about the other smarticles it’s supposed to coordinate with is what it can learn from bumping into its immediate neighbors.

“Imagine one person at a rock concert with his eyes closed,” said Joshua Daymude, a graduate student in computer science at Arizona State University who works on the smarticles project. One strategy would be to appoint a leader that orchestrates the swarm, but that approach is vulnerable to disruption — if the leader goes down, the whole swarm goes down. Another is to give each robot in the swarm a unique job to perform, but that’s impractical to implement on a large scale. “Individually programming 1,000 robots is basically an impossible task,” said Jeff Dusek, a researcher at Olin College of Engineering and a former member of the Self-Organizing Systems research group at Harvard, where he worked on underwater robot swarms. But “if every agent is following the same set of rules, your code is exactly the same whether you have 10 or 1,000 or 10,000 agents.”

An algorithm used to program a swarm has two properties. First, it’s distributed, meaning it runs separately on each individual particle in the system (the way each army ant carries out the same simple set of instructions based on whatever it senses about its local environment). Second, it incorporates randomness. This means that if an army ant senses, say, five other army ants right around it, maybe there’s a 20 percent chance it moves to the left and an 80 percent chance it moves to the right. Randomized algorithms contrast with deterministic algorithms, where each step is completely determined by what came before it.

Randomness might seem undesirable in an algorithm — after all, when you implement a procedure you’d typically like certainty about the outcome. But randomness also conveys some surprising performance advantages that, among other things, make randomized algorithms well-suited for application to smarticle swarms.
Random Guarantees

In 2015, Goldman and Randall were discussing the possibility of finding rules that would lead Goldman’s smarticles to act coherently as a group. Randall realized that the swarm behaviors Goldman was after were similar to the behavior of idealized particle systems studied in computer science.

“I was like, ‘I know exactly what’s going on,’” Randall said.

For Randall, the smarticles’ behavior resembled emergent phenomena modeled by computer scientists in many other contexts. One of the most famous examples is how segregated neighborhoods form. In the late 1960s, the economist Thomas Schelling wanted to understand how housing segregation takes hold in the absence of any centralized power sorting people into neighborhoods by skin color. He imagined a hypothetical person looking at his neighbors and deciding to move elsewhere based on how many of them looked like him. When the person moved, Schelling transported him to a random spot in the housing grid where he repeated the algorithmic process of observing his neighbors and deciding whether to stay or go. Schelling discovered that, according to his rules, residential segregation is virtually guaranteed to take hold, even if individuals prefer to live in diverse neighborhoods.

Ben Rollins for Quanta Magazine

William Savoie, a doctoral student in Goldman’s lab.
Randall realized that smarticles in a swarm resemble people in the Schelling model. In both cases, individuals have to make decisions without knowing their global position (they only know what they can see around them). And in Schelling’s model the decisions can be made with an element of randomness — if your neighbors look different from you, maybe there’s a high probability you move, but also some small probability you choose to stay put.

In 2016, Randall and her collaborators published a paper that imagined idealized particles living on a grid and deciding whether to move or stay put based on how many other particles they observed around them. The decision making was probabilistic — particles would essentially “roll” a weighted die every time they had to make a choice. Randall and her co-authors proved that if they weighted the die correctly, they were guaranteed to end up with a compressed swarm (in the same way Schelling could have proved that if he set individuals’ tolerance for diversity at the right level, segregation was unavoidable). By adjusting parameters in the algorithm, they could also guarantee that a particle swarm would move to an expanded state.

The randomness in the algorithm helps particles in a swarm avoid getting stuck in locally compressed states, where lots of isolated subgroups are clustered together but the swarm as a whole isn’t compressed. The randomness ensures that if smarticles end up in small compressed groups, there’s a chance individuals will still decide to move to a new location, keeping the process alive until an overall compressed state is reached. (It takes just a little randomness to nudge particles out of locally compressed states — it takes a lot more to nudge them out of a globally compressed state.)

Into the World

Proving that particles in a theoretical world can run a simple algorithm and achieve specific swarm behaviors is one thing. Actually implementing the algorithm in cheap, faulty, real-life smarticles clacking around in a box is another.

“Our theory collaborators are coming up with ways to program these things, but we’re just in the beginning and we can’t yet say these schemes have been transferred directly,” Goldman said.

One problem has been to get the smarticles to move as a group. At first, if the researchers packed smarticles into a confined space, the ensemble would just stagger around randomly.

But one day the physicists were observing this chaotic motion when the battery died in one of the smarticles. Goldman and his collaborators noticed that the swarm suddenly started moving in the direction of the inactive unit. The researchers reported this inadvertent finding to their computer science collaborators, who seized on the clue. The work led to the recent development of the algorithm that will always get an idealized swarm to move in a specified direction.

Little by little, the computer science experiments and the physical ones are moving closer together. The researchers hope to eventually prove theoretically that a basic algorithm, implemented in a distributed way in a large collection of small, cheap robots, is guaranteed to produce a specified swarm behavior.

“We’d like to move to a point where it’s not that batteries died and we found a phenomenon,” Daymude said. “We’d like it to be more intentional.”

This article was reprinted on ScientificAmerican.com.