

Project Swarm

How technological advances are inspired by swarming animals

Jeremy Bernstein
Manchester Grammar School
Old Hall Lane, Manchester
M13 0XT, UK

September 1, 2011

Abstract

This essay is about how animals swarm together and their impact on technology. There is a lot to learn from how groups of simple individuals can combine to form complex and intricate systems. The essay will try to find out how this swarm behaviour can influence and inspire technology, and whether or not there is scope for further advancement in the future.

A field called *Swarm Intelligence* has arisen in the last thirty years. Its foundations are in the study of swarming animals like birds and insects. Take an ant; on its own it seems to act randomly and without insight. But the colony as a whole tells a different story. The swarm of ants can carry out a wide variety of tasks, despite each ant being very much unaware of its own actions. Each ant follows a small set of simple rules; remarkably these rules have evolved so that organisation and skill at the colony level emerges - there is no chief ant telling the others what to do. These principles are now being used in robotics and computer algorithms to create more robust systems. If the ants don't have a command and control centre susceptible to attack, why must we?

The essay answers the question 'How are technological advances inspired by swarming animals' in four chapters.

The first chapter, *What is Swarm Intelligence?*, introduces the subject formally and familiarises the reader with what is to come. Ideas about emergence and self-organisation are discussed, as these themes will recur.

The second chapter, *Swarms in nature*, takes an in depth look at the different varieties of swarming animals. This includes description of the capabilities that come from swarming, as well as analysis of the motives for and consequences of the behaviour. The ideas in this chapter form the foundations of swarm-based technology, so consider it an important grounding. The swarm simulation that I made is also discussed.

The third chapter, *The whole greater than the sum of its parts*, looks at philosophical thoughts behind the swarm phenomena such as the role that randomness plays, a deeper take on emergence, and the clash between determinism and indeterminism.

The fourth chapter, *Applying the principles*, takes the ideas from the two preceding chapters, and shows how they might be used (or indeed are being used) to improve our own technology. Whilst the immediate impact that swarm research has had is on computer software, the prospects are still promising for robotics, space exploration and artificial intelligence.

Swarm Intelligence is a new subject and is little known. Currently it is quirky and wayward, but the forecast of where it may take us is magnetic. Given the infant state of swarm robotics, this essay must conclude that there is a large scope for further advancements in swarm-tech in the future. It's easy to imagine the solar system housing a swarm of space probes, exploring collaboratively. So this essay aims to bring the reader up to speed on both the applications and ideas behind swarm intelligence, with an eye on illuminating its future potential.

Contents

1	What is Swarm Intelligence?	2
2	Swarms in nature	4
3	The whole greater than the sum of its parts	9
4	Applying the principles	12
	Bibliography	18

Chapter 1

What is Swarm Intelligence?

“If you watch an ant try to accomplish something,” says Deborah M. Gordon, “you’ll be impressed by how inept it is.” [1] When the biologist at Stanford University made this seemingly cursory remark, she was not renouncing the subject that she has devoted her life to. The fact is that ants are a massively successful species; more than 12,500 species have been classified, and they span six continents. [2] Although an ant has a tiny brain and very limited intelligence, the colony as a whole is capable of remarkable tasks: they build intricate nests, find the shortest route to food, and in the event of flood build rafts out of their own bodies.

It is clear then that there is some form of greater intelligence which arises in large groups of more simple animals, and we call it *Swarm Intelligence*.

The subject has been studied under the guise of Swarm Intelligence only since 1989, although swarming animals have been examined for centuries. [3] The distinction is that the new field does not just comprise biologists documenting animal behaviour. Computer scientists, physicists, and the like are using the lessons learnt from the swarm behaviour in nature to solve our own problems. For example a swarm-based internet search engine has been researched, based on the idea that a human could lay a pheromone trail when searching for information on the web much like an ant does when searching for food. [4]

It is important to be clear on what we mean by swarm here, because with Swarm Intelligence we are not just talking about the mechanisms animals use to travel in large groups. In fact the field encompasses all the mechanisms that allow colonies of animals to perform tasks difficult or impossible for the individual alone, such as foraging for food. And we might leave the word animal behind, for there are no animals living inside computer programs based on Swarm Intelligence. The word agent is better suited. So a swarm is a group of agents. But there is more, because back in nature ants are not being told what to do by some kind of chief ant. There is a queen, but her function is not to give orders. What we find is that the organised behaviour of the colony results from the local interactions between the ants themselves. These interactions result from simple rules - in some cases merely ‘follow the ant in front of you’. So we say that the ants are self-organised, and a swarm is a group of self-

organised agents. By giving a set of simple rules to each agent in a group, the group becomes capable of complex tasks. This is a powerful tool, especially in computing. Instead of trying to face a difficult task head on, now we can just set out a simple set of rules for a large group of agents to abide by such that the desired result emerges from the system.

And this brings us to another word: emergence. Emergence involves local, and to some extent random, interactions between agents resulting in intelligent global behaviour emerging. (The previous sentence is worth re-reading: it is an important concept. A more detailed analysis of emergence is left to the next chapters, with examples in the following chapter.) Put simply, hundreds of simple events can add up to make something complex happen. In nature, this global behaviour is usually beyond the scope of understanding of the simple agents, so they are just doing what they are hard-wired, or 'programmed', to do. In the case of computer software the agents are directly programmed by a computer programmer, but in nature the rules must arise by different means. The complex, emergent behaviour renders a selective advantage as simple organisms are able to punch well above their weight - a swarm of ants might repair a nest very quickly for example. So animals hardwired to follow the right set of simple rules are more likely to survive, and emergence evolves.

Chapter 2

Swarms in nature

Swarm Intelligence has its roots in the natural world: that is where the first swarms were, after all. Swarming bees, shoals of fish and flocking birds all use swarms in interesting ways, but we shall start by taking a look at what ants gain from living collectively. Pay close attention to the natural mechanisms described, because in Chapter 4 they will be adapted and built on for swarm-based technology.

One remarkable thing about ants is that although no single ant is in charge, they always send the right number of foragers out of the nest. The number will vary depending on the conditions on a certain day. For example if there isn't much food to be collected then fewer ants will be required, but if the nest has been damaged then more ants are needed to go out and fix it. The question as to how they do it is tricky, but the answer turns out to be quite simple. Deborah Gordon has been researching red harvester ants (*Pogonomyrmex barbatus*) in the Arizona desert. Every morning a class of patroller ants leaves the nest. These patrollers have a distinctive hydrocarbon scent coating their bodies, and their job is to scout the area surrounding the nest for food. As soon as they find some food, they return to the nest. Thus the more food there is scattered around, the more frequently the patrollers will return to the nest. Forager ants waiting in the nest entrance are somehow provoked to leave the nest by the returning patrollers. Gordon wanted to know exactly how the foragers were stimulated to leave the nest, so one morning she captured the patrollers after they left the nest and performed an experiment. She dropped beads into the nest entrance. The first thing she found was that the foragers only left the nest as a response to contact with beads coated in the patroller scent; there was no reaction to the foragers being jostled by scentless beads, showing that they recognise the patroller scent. Next she varied the rate at which the beads were dropped. She found that if the time interval between dropping each bead was either too long or too short the foragers would not leave. The optimum rate was one bead every ten seconds, which resulted in the foragers gushing out from the nest. The explanation is this: drop the beads in too slowly (corresponding to a scarcity of food) and the foragers will stay in the nest because by the time a second bead touches an ant it will already have forgotten the first. But the forager ants are programmed to sit tight if they are jostled too much

(when the beads are dropped in too fast). This is beneficial because the usual cause of the over-contact would be the patrollers rushing back into the nest because of a danger outside, such as a predator. So the ants use a decentralised method based on local interactions between patrollers and foragers to fine tune the number of foragers leaving the nest, proving that it's possible to coordinate complex actions without having someone in charge. [1, 5]

Ants have the uncanny ability to select the shortest route from a food source to their nest. This is exhibited in the ant 'highways' seen often in the wild and sometimes in our kitchens. This is another example of how large numbers of simple agents with limited intelligence can perform remarkable feats whereas humans often struggle to find the shortest route. Take the Argentine ant (*Linepithema humile*), investigated by Jean-Louis Deneubourg of the Free University of Brussels. When this ant forages it deposits a trail of chemicals called pheromones. The ant will lay the pheromone trail on the way out to find food and then return along the same trail, meaning that when they return to the nest there will be a double-strength layer of pheromone. Other ants foraging for food will smell the pheromone trail and follow it, they themselves adding to the strength of the smell. But this situation is not yet quite adequate, because as it stands all the ants would follow one trail which would be continually reinforced in a positive feedback loop. There's a trick to stopping this loop, allowing the ants to find a short route: evaporation. The pheromone strength decreases over time. This means that if one ant takes a long route and one a short route to the same piece of food, then at the time when each ant gets back to the nest the pheromone trail marking the longer route will be weaker. This is because the ant on the long route took more time, allowing more of the pheromone to evaporate. Thus the next forager ants to leave the nest will be more likely to select the stronger smelling, shorter route. And so where humans would need to spend time pouring over a map (and making the map in the first place), ants succeed with some style - just by having a set of simple rules which govern local interactions. [6] This method holds a likeness to humans following a well-worn path in the woods, but a distinction does exist. The first human to pick a path used their intellect to try to pick a good route, and their footfalls began to erode the ground into a path. The first ant, however, acts more blindly; the ants rely on their large numbers working in concert to allow a quick route to emerge. With a lack of intellect they throw their resources at the problem, and it's an effective strategy. (This idea is elaborated in Chapter 3, concerning the random quality in swarms.)

Don't be too envious; sometimes these simple rules are the downfall of an ant. For example - what would happen if an ant took a few chance turns and ended up following its own pheromone trail? The sad news is that the ant ends up going round and round in an endless circle, continually adding to the strength of the trail, and enticing other ants in. [5] These so-called ant 'mills'¹ can spell death to large chunks of a colony, and they have echoes in computing - programs can get stuck in 'infinite loops', possibly causing your computer to crash.

¹Ant mill: <http://www.youtube.com/watch?v=mA37cb10WMU>.

Perhaps the most miraculous examples of Swarm Intelligence in nature are the honeybees (*Apis mellifera*) - in particular, when they decide to move house. This normally happens during late springtime: the colony, grown in size, gets too crowded and splits, leaving half the bees homeless. The dislodged bees settle temporarily in a sheltered location, while scouts go out to search for a new nest site. The scouts are looking for somewhere high up with a small entrance hole (a tree cavity for instance) that has enough room inside. Thomas Seeley, a biologist at Cornell University, led a team on an experiment to find out what happens next. Seeley brought several honeybee swarms to Appledore Island², off the coast of Maine in the States. The island has very few trees and no natural nest sites for the bees.

The team had marked every bee with a paint dot and a small plastic identification tag, and so they began by setting up five artificial nesting sites for the bees to choose from. Four were designed to be the wrong size, but the fifth was made just right. They released the bee swarm, and through careful observation discovered how the bees move house. Each scout scours the land until it discovers a possible nest. The bee examines the site briefly, and if it determines that the site is suitable it will return to the swarm, where the strange part happens. The scout bees perform a routine - a 'waggle dance' - to convey that they found a nest site to other scouts. The dance also contains a code, containing the directions to the site. The waggle dance³ is in a figure of eight shape and, somewhat unbelievably, the angle that the straight mid-section of the eight makes with the top of the nest tells the other bees what angle their flight should make with the sun. The length of the mid-section indicates the distance.

The scouts that watched the dance then go to check out the advertised nest site, and if they agree that it's a good one then they return to the swarm to dance for it themselves. Through Seeley's experiment and a number of similar ones, it turns out that there is a critical mass. As soon as about 15 scouts converge on one of the nest sites simultaneously, they sense that an agreement has been made, and they return to the swarm to bring them to the new nest site. Unsurprisingly, in Seeley's experiment the best sized site was settled on.

The honeybees use a distributed method to make their important nesting decision. This confers a lot of advantages. If there was one bee in charge of choosing the new site, it would need to scour the entire island to find the best site, and if it got lost the rest of the swarm would be out of luck. Using the bees' method the task of searching the island is split up, and no single bee has an overview of the whole process. This means when looking for a solution to the problem, the process is resilient to failure as a few dead bees will not affect any of the local interactions on a wider scale. This is one of the attractive sides to using swarms in software and robot design, but that will come later. Crucially, the decision-making process is an approximating one - the best nest site may not be found, but the bees will tend to find very good ones very quickly. [1, 7]

²Appledore Island: <http://www.projectswarm.net/images/appledore.jpg>.

³Waggle dance: <http://www.youtube.com/watch?v=4NtegA0QpSs>.

Another point of interest in the case of the bees is language. In Karl Popper's essay *Of Clouds and Clocks*, he discusses how language can be divided into lower and higher functions. There are two lower functions, within the reach of many animals. The first is the ability to convey a symptom by making some form of signal; a wolf in pain may squeal, for instance. The second is the ability to respond to such a signal; other wolves may come to the rescue. The two higher functions are more elaborate: description and argument. The lower functions are always present in communication, but it is usually only humans who are able to describe something or argue about what's right. [8] But now let's look at the bees. Whenever a bee performs the waggle dance, it is actually describing the whereabouts of the nest site. Impressive, but this individual bee is not capable of the other higher function - argument. Nevertheless the bees do reach a decision about the new nest site: while each bee is not capable of arguing, collectively the swarm is. This is a great illustration of how simple agents can collaborate (or *swarm*) to overcome limitations on their intelligence, and it speaks volumes for the potential of collaborative design, say, in robotics.

A superorganism is a collection of agents [which] could act in concert to produce phenomena governed by the collective. [9] Fish in a shoal form a superorganism very well adapted to avoiding predators. The more pairs of eyes in the shoal, the more alert it is to danger. When a fish sees a shark it will dart out the way; the neighbouring fish also dart out the way in response to the first fish. This domino effect causes shockwaves to ripple through the shoal, and every fish is made aware of danger staggeringly quickly. The shoal uses its swarm nature to its advantage by behaving as a superorganism. Many behave as one, and the shoal has tricks that make it extremely difficult for a predator to track any one fish. For example, it may form a bubble⁴ around the predator, or explode, sending fish in all directions. [5]

These tricks, and the superorganism itself, are all examples of the complex behaviour which emerges from a swarm. They result from the local interactions between the agents, the fish responding to their neighbours.

Birds also display such emergent behaviour, like when starlings⁵ wheel across the sky in huge flocks.

There are several reasons starlings perform these aerial acrobatics: to intimidate and bamboozle predators, to attract more starlings for greater safety in numbers, and perhaps as a form of socialising - starlings are more intelligent than ants after all. [10] But in 1986 a computer graphics researcher, Craig Reynold, was more interested in *how* the starlings flocked together than *why*, and he wanted to make a realistic computer simulation. He knew the behaviour had to emerge from each bird following simple rules, because each bird only has access to local information about other birds in its immediate vicinity. So with this in mind he sought to construct a model where each agent follows a limited number of rules. The agents in this model are called *boids* (birds in a New York accent), and they live in a three dimensional world. Reynolds

⁴Fish bubble: <http://www.youtube.com/watch?v=zvfY8-3ktNA>.

⁵Starling flock: <http://www.youtube.com/watch?v=eakKfY5aHmY>.

whittled down to only three rules necessary for each boid: first, steer away from nearby boids if you are much too close (to avoid getting in their way); second, steer towards nearby boids so long as you are not too close already; and third, steer so as to head in the same general direction as nearby boids. It turned out that a good balance of these simple rules yielded a remarkably convincing model of a flock of birds. Unpredictable, life-like movements had emerged. Observation and experiment with real starlings has confirmed that these rules are the ones they follow. [5, 11] Boids' feature film debut came as a swarm of computer-generated bats in the 1992 film 'Batman Returns'; it is still used in films and video games.

I have developed my own rendition of Reynolds' Boids, written entirely in Javascript. The app⁶ is based on the same principles: nearby agents try to stay close, but not too close, and they try to move in the same direction. I call these attraction, repulsion, and direction matching. But there are four key differences. First, my version is two dimensional, but still serves as a good representation of emergence and flocking behaviour. Second, my version is built on top of a physics engine, which is what brings real-world physics (for instance gravity) to computer simulations. What this means for my app is that the agents can collide with each other and bounce off. The third difference is that I have implemented the three rules of attraction, repulsion and direction matching in a different way: Reynolds uses the rules to affect the steering of each boid, but my app directly changes the velocities. To explain this better: imagine the boids are cars. For a boid to change direction completely in Reynolds' simulation, it must perform a U-turn. In my app the agents can stop and put the car in reverse gear, thus moving off backwards without leaving the same straight line. This is why Reynolds' boids look more like flocking birds, whereas my agents more closely resemble swarming insects (which are more manoeuvrable). The fourth difference is that my app is interactive. We do not suffer from the hardware constraints that Reynolds did in the 1980s, so I was able to add parameters which can be tinkered with to alter the animation in real-time. For example you can change the strength of attraction, the radius of repulsion (how close other agents must be before they will be repelled), and the number of agents. Furthermore, Reynolds boids had a fixed field of view (that is, they could not see behind them), whereas I have made it adjustable. 360 corresponds to being able to see in every direction, and 0 makes the boids blind. Reynolds had to deliver the finished product: you click go and you've got a flock of birds. My app on the other hand allows the user to tweak the controls in order to build their own flock, so the user garners an appreciation of how finely-tuned the rules of local interaction must be for emergent behaviour to develop. With care the agents can be made to form a convincing bird flock.

If you were sceptical of the principle of emergence, the app should change your mind. It is a manifestation of complex behaviour emerging from simple rules - and in this case, we made the rules!

⁶The app can be used here: <http://www.projectswarm.net/predator/predator.html>. I updated the app on 15.10.11 to include the option of a predator.

Chapter 3

The whole greater than the sum of its parts

The idea of Swarm Intelligence is based on an interesting proposition. It is that by combining large numbers of simple things, we can make something of which the complexity outweighs that of the uncombined components. This should not be a new concept: a large number of silicon switches and bits of metal can be combined to make an iPhone. Even the bricks that make your house are arranged in such a way that they stop being bricks and start being a place to live, or perhaps a home. More fundamentally, everything in our world is made of atoms. But there is a distinction between a whole swarm and a whole house. If you looked at a house, you would not be shocked to realise that it stood up because of the interactions between the bricks - that is, one brick sits on top of another - and that the house's construction stems from human intention. But you may be more surprised to learn that the construction of a termite mound comes as a result of the local interactions between unthinking termites. The termites do not collectively decide to build it, and they are oblivious to the fact that it is being built. For this reason we say that the mound-constructing behaviour of the termites *emerges* from the local interactions of the agents; a house, on the other hand, is just a house. This emergent behaviour is fascinating because the swarm as a whole can have complex aims and actions, but they rely entirely on the agents which are clueless. It would be hard to anticipate the behaviours by looking at the agents. As with the example of the fish-shoal-superorganism: each fish only knows that it doesn't like sharks and that it should jolt off when its neighbour does; the collective swarm knows how to dazzle and confuse the sharks for the good of every fish.

The idea that a whole can be greater than the sum of its parts is sometimes referred to as holism. [12] Holism's main focus is that in order to understand a complex system you must look at the system itself and not its constituent parts. This is counter to reductionism, which is the way that scientists normally look at things. Reductionism involves breaking the system down into smaller and smaller parts to gain

a greater understanding of the whole. [13] Reductionism certainly has its merits, as the overwhelming majority of the science we know today can be attributed to it. But as we dismantle systems that are more and more complex, it gets harder and harder to work out how the building blocks piece together to form the functioning whole. In these cases, a holistic approach can be useful - for example ecological scientists interested in the natural world study ecosystems rather than looking for smaller and smaller chunks.

We have decided in a practical sense that reductionism is good if you can use it; otherwise we can turn to holism. But it remains to be seen which is more right. Going back to our example of flocking birds, there is a question. If we had complete knowledge of one bird, could we predict the precise patterns and behaviours the flock would display when wheeling around the sky? Reductionism says yes; holism says no. This relates to a bigger question of determinism versus indeterminism in science. A deterministic system is such that if everything about its current state is known, its entire history is deducible and its entire future is predictable. Karl R. Popper referred to such systems as perfect clocks, reflecting our tendency to say that something regular and reliable is running like clockwork. On the other hand, an indeterministic system is one where total knowledge about the present is not enough to ascertain the past or predict the future. There are some inherently unpredictable elements in such systems, and with reference to the vagaries of the weather, Popper called them clouds. [8] We can thus form a new question: if we record the movements of a flock of birds, and then return the flock to exactly the same initial conditions (in an ideal world where this is possible), will the flock move in an identical manner the second time around?

In the time after the Newtonian revolution, the universe became widely held as one big clock. If all interactions were governed by Newton's laws of nature, how could anything unpredictable ever happen? I am not going to go into quantum physics here, but it took until the quantum revolution in the twentieth century before scientists and philosophers widely began to recognise that chance is built into the fabric of the world. When it was discovered that physical interactions occur according to statistical probabilities, we realised that the world must be indeterministic. And so we have an answer to our question. The movements of the flock and its replica would be different. It follows that given complete knowledge of a single bird we could not predict the precise movements of a flock. This is because at some level chance is involved. In the short term, we can make a good approximation of the position of the flock. But in the long term the chance effects are so amplified that we cannot tell where each bird will be or in what direction it will move. It is interesting to note that Popper's prime example of a cloud is a swarm of gnats. The flock is on some level a *chaotic* system, meaning that small differences in conditions (perhaps due to chance - though this is not strictly chaos theory) lead to wildly different outcomes in the long run. [14] Therefore a reductionist attitude cannot tell us about the position of the flock, and we might have to look to holism.

This sets up a hurdle when designing artificial swarm systems. A holistic approach

is useful for analysing the movements of swarms: for example if a lot of starlings were thirsty we could deduce that they would settle by a lake, or we could predict the ‘bubble’ shape that a shoal of fish would form around a predator. But when creating our own nature-inspired system for a certain task we need the desired behaviour to emerge, so we need the right set of rules governing local interactions. The holistic approach gives little help in finding these rules; a blind trial and error approach would take a very long time. This explains why scientists designing systems inspired by natural swarm phenomena rely on careful observation of the local interactions in the original biological system. These interactions have been perfected by natural selection over the millennia, so they provide a good head start for humans designing the artificial interactions.

Chance, or randomness rather, plays another important role in the world of swarms. Let’s go back to the earlier example of ants finding the shortest route by laying pheromone trails which evaporate. Take the case where two ants take two different routes: one long and one short. If the next ants to arrive on the scene strictly only followed pheromone trails already laid, then they would have to choose between these two routes depending on the strength of the pheromone, and a third, shorter route might be missed. Moreover, ants would never stray from the pheromone trail they were on, so the routes would never be improved. This would be problematic because very short routes would rarely be found. But in the real world a whole number of factors work to avoid this problem - random factors. For example part of a pheromone trail might be washed away by a drop of water, or an ant (with a blocked nose) may fail to smell that a pheromone trail is there. Factors of this sort mean that whilst most ants do follow the trails set out, others will randomly stray off the beaten track and make new paths. Of course these paths might be very long, but then the pheromone will evaporate and they will be forgotten. Chance might have an ant strike lucky. It is interesting to see that swarms rely on their random character to find optimal solutions to problems. You’d be surprised if a human, late for work, stopped to roll a die at every crossroads in the hope of getting there faster. But swarms are of a different nature; each agent is simple and expendable, and for the ants the best way of doing things may seem counter-intuitive to us.

Chapter 4

Applying the principles

Swarms are not just an interesting phenomenon in nature. We can use the ideas learnt from studying natural systems like ant colonies to develop and improve our own technological systems. The general swarm principle is that from simple local interactions, complex behaviour can emerge. Using this principle to design a technological system can make things much simpler, as we don't need to deal directly with the complexity of the system as a whole. Other advantages are that the system is distributed so when individual components fail the system can survive, and the system is self-organising so there is no leader to be relied on.

So far the most important application of Swarm Intelligence has been in optimisation algorithms. Combinatorial optimisation is used to find the best element in a set for some purpose. It is extremely useful in the real world, as when Google Maps is trying to find the shortest route from one postcode to another. The Travelling Salesman Problem is often used as a touchstone for optimisation algorithms. The problem is this: given a set of cities and the distance between every pair of cities, what is the shortest path that visits each city exactly once? It is by no means easy to solve, and the number of possible routes increases faster than exponentially with the number of cities. For n cities, there are $n!$ route possibilities, meaning for 15 cities there are more than a trillion routes. Interestingly a number of other complex problems like assembling DNA sequences and designing silicon chips reduce to a modified version of the Travelling Salesman Problem, making it a very lucrative problem to solve efficiently. Dr Marco Dorigo, one of the founders of the Swarm Intelligence field, made a link between this problem and that uncanny ability of ants to find the shortest route from their nest to food. Ant Colony Optimisation was born. It's important to stress that this is a heuristic technique, meaning that it quickly gets satisfactory solutions though they may not be optimal; this is usually preferable because for larger numbers of cities getting the optimal solution by brute force (finding the length of every possibility and comparing to see which is shortest) becomes intractable.

It works like this: virtual ants are dispersed at random cities and each makes a tour, visiting every city once. Nearer cities are favoured when deciding which to visit next, but otherwise the decision is random. When the tour is complete, virtual pheromone

is deposited along the route. The thickness splashed down is inversely proportional to the total length of the tour, so shorter tours get more pheromone. The ants then make another tour, but this time ants favour paths with thicker pheromone trails as well as nearer cities. As with the real-life foraging ants, pheromone evaporates to avoid the process settling prematurely on poor solutions. Dorigo found that when this process is iterated, near-optimal solutions are found. There is a specific advantage to Ant Colony Optimisation over competing heuristic techniques: flexibility. The algorithm can be run continuously, the ants just keep going round exploring different paths, and they can respond to changes in real time. If there's been an accident and a road is closed between two cities, then backup plans already exist in the virtual pheromone trails, and the ants start to select different routes from an effective pool of alternatives. [6, 15]

A number of companies now use variants of Ant Colony Optimisation to run things more efficiently. For example, 'American Air Liquide' uses an algorithm to route trucks delivering gas from plants to customers. In the past the drivers had simply collected gas at the plant closest to the customer, but now they were instructed to drive to the plant which would result in the cheapest delivered price (as prices fluctuate and vary from plant to plant). It didn't make sense to the drivers, who were sometimes being asked to drive much further afield, but the company reported big savings. [5]

The flexibility of ant-based methods is also attractive to communications companies. Take telephone networks: conditions change unpredictably, transient surges of traffic occur during TV phone-in competitions, local switching stations are swamped during pop concerts. When a phone call is routed, it traverses switching stations, or nodes, on its way to the other end. There is a routing table at each node directing calls to another node depending on their ultimate destination. Calls should avoid busier nodes to reduce the strain on the network and speed up the user's connection. Ant-based approaches to the problem have been developed. One approach, designed by researchers at the University of the West of England and HP's labs in Bristol, involves ant-like agents spreading through the network. When they travel from one node to another, they alter the routing table score for that pair of nodes; this is the pheromone's analogue. A fast journey means the score is greatly increased, but a slow one only adds a little. Evaporation is implemented by the routing table entries diminishing over time. The weightings of evaporation and reinforcement are such that a slow, busy route with more agents receives a lower score than a fast route with fewer agents. Phone calls favour paths between nodes with higher scores. This provides a flexible system where a previously fast route that gets congested will quickly be dropped in favour of faster alternatives. British Telecom have applied an ant-based routing technique to their telephone network, but the notoriously unpredictable internet may be the ultimate frontier. Dorigo and another researcher Gianni Di Caro have developed 'AntNet' for this purpose. It is similar to the routing method just described, but with some improvements. Packets of information hop from node to node, and they leave a trace indicating the speed of that packet's entire journey, not

just the journey speed from the first node to the second. Other packets favour paths with stronger traces. In tests AntNet has outperformed all existing routing protocols, including the internet's current protocol 'Open Shortest Path First'. AntNet is better at adapting to changes in the volume of traffic and has greater resistance to node failures. Routing companies have shown interest in AntNet, but its use would require replacing current hardware at huge expense. [6, 15]

Following on from the ant algorithms, the swarm principle has also been used to develop a new kind of internet search engine. Search queries fall into three categories: navigational, perhaps to visit Facebook; informational, to find out who invented the light bulb; and transactional, to buy something off eBay. Two Spanish researchers sought to build a search engine that yields a greater proportion of search results relevant to the user's query than the current leading search engines. It is swarm-based, but this project departs from the Swarm Intelligence mainstream as virtual agents are not employed; we, the humans, are agents. The backbone of their idea is that when a human searches for something on the web, it is analogous to an ant foraging for food. The piece of food is the chunk of information, so it is called information foraging. Wouldn't it be easier for a second person to find the same chunk of information if the first laid a pheromone trail? This method seems less wasteful than the current method of users browsing page snippets and clicking links until they find what they are looking for (or indeed give up). Underpinning this process is the idea that when a human clicks a link, they are issuing a 'relevance judgement': they are deeming that website relevant to their query.

In this swarm model, the path from the query to the clicked page is marked with pheromone. The more pheromone deposited on a path, the higher up the search listings that page will appear. Pheromone evaporates, like in other swarm-based systems, to prevent mediocre query-to-webpage paths being settled on. The researchers also use a stochastic mechanism to prevent initially popular pages becoming too dominant. There is a risk that a webpage listed at the top of the search results would be reinforced disproportionately to its relevance, and a very relevant webpage on the second page of results might go unnoticed. To avoid this, a random nature is built into the model, whereby heavily-hit results are more likely to appear at the top of the search listings, but less-visited sites can still appear by chance, be clicked on, and thus reinforced. This swarm-based method has a key advantage over other similar techniques that 'learn' relevance from user clicks: it works in real time, so it can rapidly respond to the changes in trends of what people are after; other techniques need to be retrained off new data that's been recorded rather than adapting on the fly. Evaporation also incorporates the decay of user interest in a natural way. The researchers tested their search engine with real users, using Yahoo as a control, and the results were promising. Successive users completed tasks successively faster using the swarm engine, as the search engine learnt from the experience of previous users. Later on when it was more trained, the users of the swarm engine also tended to complete harder tasks considerably faster than the users of Yahoo. The swarm principle is used

here to enhance an activity that people carry out every day. The researchers point out a limitation: certain results under certain queries could be boosted by a malicious ‘swarm’ of users. But Google has had similar troubles and some form of detection system could be designed to alleviate the problem. This swarm-based search engine exemplifies the diverse set of applications that the swarm principle offers. [4]

The most clear-cut application of the swarm principle in technology is swarm robotics. Dorigo turned his attention from swarm algorithms to a ‘Swarmanoid’¹ project. The idea is that a swarm of small, cheap robots can cooperate to perform tasks more effectively than one larger, more expensive robot. In the swarm there are three kinds of robot: hand-bots, foot-bots, and eye-bots. The eye-bots fly to explore an environment and locate objects of interest. The hand-bots, capable of climbing, are carried to the objects by the foot-bots and pick them up. The robots use light signals to convey information to their neighbours; a red light might indicate that an object has been found, for instance. This system is robust, because if one robot breaks the others carry on unhindered. It is also flexible and scalable (it can work with five robots or fifty), because it relies entirely on local interactions between the robots and there is no central coordination. In principle the robots are suited to exploring unfamiliar environments, such as a burning building; Dorigo says that in a more advanced state the robot system could be used to rescue people or possessions. [15] It is important to stress that we live in the very early days of swarm robotics, and it still needs a lot of work to get it further off the ground. But this leads me to think that it has a lot of potential for the future, and is the main reason I would argue that the entire Swarm Intelligence field has great future potential.

Currently we do not know how human consciousness arises, but it’s a nice idea that it might emerge from the local interactions between neuron cells firing in the brain. Dr Vito Trianni, of the Institute of Cognitive Sciences and Technologies in Rome, is a proponent of ‘swarm cognition’, and suggests that this might be where consciousness comes from. [15] In a human brain there are an estimated 90 billion neurons and a quadrillion synaptic connections (that’s 10^{15}). [16] We could liken the brain to a swarm and the neurons the agents. A team at Manchester university led by Professor Steve Furber is basing the design for a new kind of computer on the structure of the human brain. The computer is called SpiNNaker - a contraction of ‘spiking neural network architecture’. Traditional computers work by completing a sequence of operations one after the other. Computers have been made with dual core processors so two sequences of operations can be completed at the same time, and the operations occur in step with the system clock. More expensive four or even six core processors have become available recently. Generally more cores means higher performance, and this is called parallel processing. But the human brain works in a different way: neurons fire throughout the brain when they are stimulated, and signals cascade through the brain. The neurons do not fire in step with a ‘system clock’. Amid the many billions of brain cells functioning simultaneously, there are not

¹Project Swarmanoid: <http://www.youtube.com/watch?v=M2nn1X9X1ps>.

just one, two, four or six sequences of operations occurring at once. The brain also has a high level of redundancy: getting hit on the head may cause the loss of countless neurons, but probably won't spell death for a human.

Furber's team designed a chip with 18 cores for the SpiNNaker project, and each chip can be linked to six other chips facilitating a massively parallel network. The cores send packets of data to each other to mimic the signals sent from neurons as electrical impulses. The chips are said to be *asynchronous*, because they are not governed by a global clock signal; instead the cores simply act when interacted with, facilitated by handshake signals being made when one core wants to talk to another. This is reminiscent of swarms not relying on top-down control but local interactions. Being asynchronous draws the computer closer to how the brain works, and has other advantages such as using much less power as cores can shutdown when not being interacted with. The network is also designed to have redundancy; if a chip is broken signals will be routed around it. The end goal is to build a computer incorporating one million of these chips. This computer should have 1% of the power of the human brain. The purpose of the computer will be to help understand how the brain processes information. The team does not claim to be building some kind of Frankenstein-robot-brain, or to understand the human brain, but they hope that SpiNNaker will be useful as a tool for medics and scientists when studying complex brain injuries and diseases; it will be a platform for studying the flow of information in a complex system in many ways akin to the brain. It's still an exciting thought that some form of intelligence might emerge from the machine. [17, 18, 19]

Swarm-based methods prove to have certain advantages when it comes to designing technological systems. Their distributed nature makes them more robust and fault-tolerant as they lack a single weak point. Their self-organised nature provides flexibility to deal rapidly with fluctuating, unpredictable and perhaps unexpected situations. Swarm Intelligence gives us a way of designing systems that have intrinsic autonomy, without need for extensive pre-programming and central control. Swarm Intelligence is essentially a form of artificial intelligence; it's nice to have a system that will operate by itself, giving us the results without too much human oversight. But there are disadvantages: in nature's ant mills the ants die because they are unable to realise that they are stuck in a loop. We can imagine a swarm of robots with no supervisor getting stuck in similar pickles. Whilst being apt for optimisation tasks, swarm-based techniques aren't always suited to tasks requiring a greater depth of reasoning. Another criticism of the field is that using simple agents that display somewhat random behaviour could result in unpredictable behaviour and inefficiencies. [6] This point of view, however, is probably more of a misunderstanding of the swarm principle. Although at the agent level there is randomness, the intention is for that to disappear at the swarm level when the desired behaviour emerges. Furthermore, the random quality provides the inherent flexibility of the swarm. It could be looked at as a great asset, potentially allowing swarm systems to deal with unforeseen problems, an ability absent from traditional software.

At the beginning of this essay, we sought to find the answer to two questions. How do swarming animals inspire technological advances? And, is there greater scope for Swarm Intelligence in the future? In answer to the first question, this essay has described in depth how technology is derived from swarms, and analysed the current manifestations. The second question has been answered as well as can be: the entire field of robotics is in its infancy, so we could consider swarm robotics to be scarcely out of the womb, and with much room to grow. The European Space Agency is currently researching swarm robotics as a means of space exploration. So we can see Swarm Intelligence holds considerable promise for the future. Whilst artificial intelligence and robotics continue to advance, the swarm is something to watch out for.

Bibliography

- [1] Peter Miller. *Smart Swarm*. Collins, 2010.
- [2] Hymenoptera Name Server. Number of species recorded in formicidae. http://osuc.biosci.ohio-state.edu/hymenoptera/tsa.sppcount?the_taxon=Formicidae, August 2011.
- [3] Gerardo Beni and Jing Wang. Swarm intelligence in cellular robotic systems. volume 102, Tuscany, Italy, 1989. Proceedings of NATO Advanced Workshop on Robots and Biological Systems.
- [4] Daniel Gayo-Avello and David Brenes. Making the road by searching - a search engine based on swarm information foraging. *Computing Research Repository*, abs/0911.3, 2009.
- [5] Peter Miller. The genius of swarms. *National Geographic Magazine*, 2007.
- [6] Eric Bonabeau and Guy Théraulaz. Swarm smarts. *Scientific American*, 2000.
- [7] T.D. Seeley, P.K. Visscher, and K.M. Passino. Group decision making in honey bee swarms. *American Scientist*, 94:220–229, 2006.
- [8] Karl R. Popper. *Objective Knowledge: An Evolutionary Approach*, chapter Of Clouds and Clocks. Oxford, revised edition edition, 1979.
- [9] Kevin Kelly. *Out of control: the new biology of machines, social systems and the economic world*, page 98. Addison-Wesley, 1994.
- [10] The Royal Society for the Protection of Birds. Starling flocks: a wild spectacle. <http://www.rspb.org.uk/wildlife/birdguide/name/s/starling/roosting.aspx>.
- [11] Craig Reynolds. Boids. <http://www.red3d.com/cwr/boids/>, 2001.
- [12] Nicholas A. Christakis. World question center. http://www.edge.org/q2011/q11_6.html.
- [13] John C. Polkinghorne. Reductionism. <http://www.disf.org/en/Voci/104.asp>.

- [14] Stephen H. Kellert. *In the Wake of Chaos: Unpredictable Order in Dynamical Systems*. University of Chicago Press, 1993.
- [15] Riders on a swarm. *The Economist*, August 2010.
- [16] Frederico Azevedo. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *The Journal of Comparative Neurology*, 513:532–541, 2009.
- [17] Nicole Kobie. One million arm chips to mimic 1% of the brain. <http://www.pcpro.co.uk/news/368536/one-million-arm-chips-to-mimic-1-of-the-brain>.
- [18] The University of Manchester. Spinnaker - a universal spiking neural network architecture. <http://apt.cs.man.ac.uk/projects/SpiNNaker/>.
- [19] Electronics Weekly. One million arm cores to simulate brain at manchester. <http://www.electronicweekly.com/Articles/2011/07/08/51421/one-million-arm-cores-to-simulate-brain-at-manchester.htm>.