

Please read Sections 1, 2.1, 2.1.1, 2.3, 3.1, 3.1.1, and 3.2.
You can skip Sections 2.1.2, 2.1.3, 2.2, 3.1.2, 3.2.1, and 4

Swarm Intelligence

David Corne, Alan Reynolds and Eric Bonabeau

Increasing numbers of books, websites and articles are devoted to the concept of ‘swarm intelligence’. Meanwhile, a perhaps confusing variety of computational techniques are seen to be associated with this term, such as ‘agents’, ‘emergence’, ‘boids’, ‘ant colony optimisation’, and so forth. In this chapter we attempt to clarify the concept of swarm intelligence and its associations, and we attempt to provide a perspective on its inspirations, history, and current state. We focus on the most popular and successful algorithms that are associated with swarm intelligence, namely ant colony optimisation, particle swarm optimisation, and (more recently) foraging algorithms, and we cover the sources of natural inspiration with these foci in mind. We then round off the chapter with a brief review of current trends.

1, Introduction

Nature provides inspiration to computer scientists in many ways. One source of such inspiration is the way in which natural organisms behave when they are in groups. Consider a swarm of ants, a swarm of bees, a colony of bacteria, or a flock of starlings. In these cases and in many more, biologists have told us (and we have often seen for ourselves) that the group of individuals itself exhibits behaviour that the individual members do not, or cannot. In other words, if we consider the group itself as an individual – the *swarm* – in some ways, at least, the swarm seems to be more intelligent than any of the individuals within it.

This observation is the seed for a cloud of concepts and algorithms, some of which have become associated with swarm intelligence. Indeed, it turns out that swarm intelligence is only closely associated with a small portion of this cloud. If we search nature for scenarios in which a collection of agents exhibits behaviour that the individuals do not (or cannot), it is easy to find entire and vast sub-areas of science, especially in the bio-sciences. For example, any biological organism seems to exemplify this concept, when we consider the individual organism as the ‘swarm’, and its cellular components as the agents.

We might consider brains, and nervous systems in general, as a supreme exemplar of this concept, when individual neurons are considered as the agents. Or we might zoom in on certain inhomogeneous sets of bio-molecules as our ‘agents’, and herald gene transcription, say, as an example of swarm behaviour. Fortunately, for the sake of this chapter's brevity and depth, it turns out that the

swarm intelligence literature has come to refer to a small and rather specific set of observations and associated algorithms. This is not to say that computer scientists are uninspired by the totality of nature's wonders that exhibit such 'more than the sum of the parts' behaviour – much of this volume makes it clear that this is not so at all. However, if we focus on the specific concept of swarm intelligence and attempt to define it intensionally, the result might be thus:

1. Useful behaviour that emerges from the cooperative efforts of a group of individual agents;
2. ... in which the individual agents are largely homogeneous;
3. ... in which the individual agents act asynchronously in parallel;
4. ... in which there is little or no centralised control;
5. ... in which communication between agents is largely effected by some form of stigmergy;
6. ... in which the 'useful behaviour' is relatively simple (finding a good place for food, or building a nest -- not writing a symphony, or surviving for many years in a dynamic environment).

So, swarm intelligence is not about how collections of cells yield brains (which falls foul of at least items 2, 5, and 6), and it is not about how individuals form civilizations (violating mainly items 3, 5 and 6), and it is not about such things as the lifecycle of the slime mould (item 6). However, it is about individuals co-operating (knowingly or not) to achieve a definite goal. Such as, ants finding the shortest path between their nest and a good source of food, or bees finding the best sources of nectar within the range of their hive. These and similar natural processes have led directly to families of algorithms that have proved to be very substantial contributions to the sciences of computational optimisation and machine learning.

So, originally inspired, respectively, by certain natural behaviours of swarms of ants, and flocks of birds, the backbone of swarm intelligence research is built mainly upon two families of algorithms: ant colony optimisation, and particle swarm optimisation. Seminal works on ant colony optimisation were Dorigo et al (1991) and Coloni et al (1992a; 1992b), and particle swarm optimisation harks back to Kennedy & Eberhart (1995). More recently, alternative inspirations have led to new algorithms that are becoming accepted under the swarm intelligence umbrella; among these are search strategies inspired by bee swarm behaviour, bacterial foraging, and the way that ants manage to cluster and sort items. Notably, this latter behaviour is explored algorithmically in a subfield known as swarm robotics. Meanwhile, the way in which insect colonies collectively build complex and functional constructions is a very intriguing study that continues to be carried out in the swarm intelligence arena. Finally, another field that is often considered in the swarm intelligence community is the synchronised movement of swarms; in particular, the problem of defining simple rules for individual behaviour that led to realistic and natural behaviour in a simulated swarm. 'Reynolds' rules' provided a

general solution to this problem in 1987, and this can be considered an early triumph for swarm intelligence, which has been exploited much in the film and entertainment industries.

In the remainder we expand on each of these matters. We start in section 2 with an account of the natural behaviours that have inspired the main swarm intelligence algorithms. Section 3 then discusses the more prominent algorithms that have been inspired by the techniques in section 2, and section 4 notes some current trends and developments and offers some concluding remarks.

2. Inspiration from Nature

2.1 Social Insects and Stigmergy

Ants, termites and bees, among many other insect species, are known to have a complex social structure. Swarm behaviour is one of several emergent properties of colonies of such so-called *social insects*. A ubiquitous characteristic that we see again and again in such scenarios is *stigmergy*. Stigmergy is the name for the *indirect* communication that seems to underpin cooperation among social insects (as well as between cells, or between arbitrary entities, so long as the communication is *indirect*).

The term was introduced by Pierre-Paul Grassé in the late 1950s. Quite simply, stigmergy means communication via signs or cues placed in the environment by one entity, which affect the behaviour of other entities who encounter them. Stigmergy was originally defined by Grassé in his research on the construction of termite nests. Figure 1 shows a simplified schematic of a termite nest. We will say more about termite nests in section 2.1.3, but for now it suffices to point out that these can be huge structures, several metres high, constructed largely from mud and from the saliva of termite workers. Naturally, the complexity and functionality of the structure is quite astounding, given what we understand to be the cognitive capabilities of a single termite.

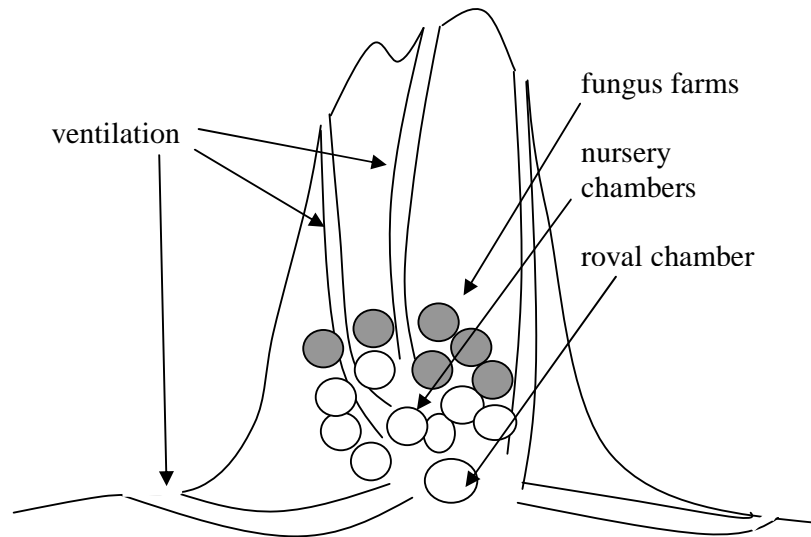


Figure 1. A highly simplified schematic of a termite nest

Following several field trips to Africa in the late 1930s and 1940s studying termites and their nests, among other things, Grasse showed that the regulation and the co-ordination of the nest-building activity did not depend on the termite workers themselves, but was instead achieved by the nest itself. That is, some kind of stimulating configuration of materials triggers a response in a termite worker, where that response transforms the configuration into another configuration that may, in turn, trigger yet another, possibly different, action performed by the same termite or by any other termite worker in the colony. This concept of stigmergy was attractive and stimulating, but at the time, and often today, it was and is often overlooked by students of social insects because it leaves open the important operational issue of how the specific trigger-response configurations and stimuli must be organized in time and space to allow appropriate co-ordination. But despite the general vagueness of Grasse's formulation, stigmergy is recognised as a very profound concept, the consequences of which are still to be fully explored. Stigmergy is not only of potential importance for our understanding of the evolution and maintenance of social behaviour in animals, from communally breeding species to highly social insects, it is also turning out to be a crucial concept in other fields, such as artificial intelligence, robotics, or the social, political and economic sciences. Meanwhile, in the arena of natural computing,

stigmergy is the fundamental concept behind one of the main swarm intelligence algorithms, as well as several others.

Apart from termite nests, another exemplary case of stigmergy in nature is that of pheromone deposition. Ants deposit pheromone along their paths as they travel; an ant striking out on her own will detect pheromone trails, and prefer to follow such trails already travelled. In general, the concept of stigmergy captures underlying commonalities in (usually) insect behaviours that are underpinned by indirect communication. This covers more emergent behaviours than trail-following, and (the original inspiration for the term) the construction of structures such as termite mounds and bee hives. Stigmergy also seems key to behaviours such as brood sorting and cemetery clustering -- some ant species are known to spatially cluster their young into age-groups within the nest, and they keep their nests tidy by removing dead nest mates and piling them into clusters outside.

The phenomenon of stigmergy has much earlier evolutionary roots; it is now used to explain the morphologies of multicellular organisms, sea-shells, and so forth. Essentially, individual cells position themselves in a way influenced by deposits left behind by their colleagues or precursors. A useful way to think of it is that stigmergic communication involves a 'stigmergy structure' which is like a notepad, or an actual structure, built from cues left by individuals.

The structure itself may be a spatially distributed accumulation of pheromone, or a partially built hive, or a partially constructed extracellular matrix. The structure itself influences the behaviours of the individuals that 'read' it, and these individuals usually also add to the structure. Army ants find their directions of travel influenced by pheromone trails, and they add to the trails themselves. Termites are triggered by particular patterns that they see locally in the partially built mound, and act in simple and specific ways as a result, resulting in additions to the structure itself. An authoritative overview of stigmergy associated behaviours in nature is Bonabeau et al (1997), while Theraulaz (1994) provides a comprehensive survey of self-organisation processes in insect colonies. As hinted above, when we consider the stigmergic processes often observed in nature, the most prominent sources of inspiration from the swarm intelligence viewpoint are those of navigation to food sources, sorting/clustering, and the collective building of structures. We briefly consider each of these next.

2.1.1 Natural Navigation

Navigation to food sources seems to depend on the deposition of pheromone by individual ants. In the natural environment, the initial behaviours of a colony of ants in seeking a new food source is for individual ants to wander randomly. When an ant happens to find a suitable food source they will return to their colony; throughout, the individual ants have been laying pheromone trails. Subsequent ants setting out to seek food will sense the pheromone laid down by their precursors, and this will influence the path they take up. Over time, of course, pheromone trails evaporate. However, consider what happens in the case

of a particularly close food source (or, alternatively, a faster or safer route to a food source). The first ant to find this source will return relatively quickly. Other ants that take this route will also return relatively quickly, so that the best routes will enjoy a greater frequency of pheromone laying over time, becoming strongly fancied by other ants. The overall collective behaviour amounts to finding the best path to a nearby food source, and there is enough stochasticity in the process to avoid convergence to poor local optima – trail evaporation ensures that suboptimal paths discovered early are not converged upon too quickly, while individual ants maintain stochasticity in their choices, being influenced by but not enslaved by the strongest pheromone trail they sense. Figure 2 shows a simple illustration, indicating how ants will converge via stigmergy towards a safer and faster way to cross a flow of water between their nest and a food source.

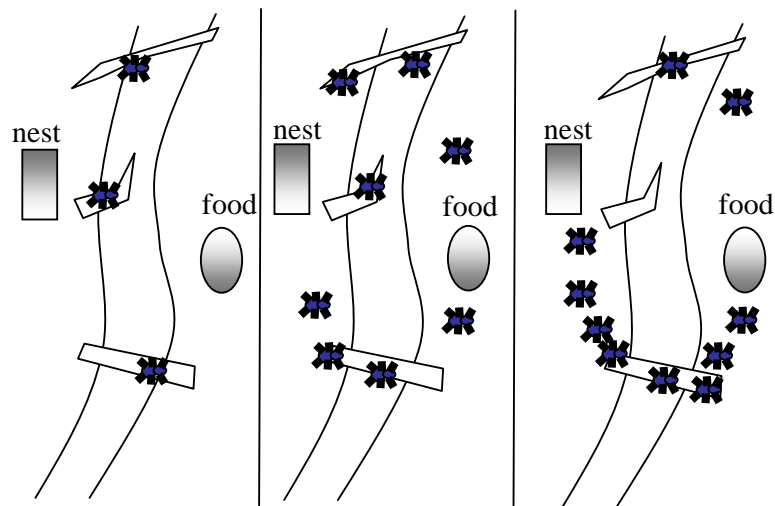


Figure 2: Convergence to a safer crossing over time.

In figure 2, we see three contrived snapshots of a simple scenario over time. On the left, ants need to cross a narrow stream of flowing water towards a tempting food source, and three crossings – fallen twigs – present themselves. Initially ants are equally likely to try each one. Each of these ants lays its trail of pheromone as it makes its journey towards the right. Over time, however, the path

towards the middle twig becomes less laid with pheromone, simply because, unlike the situation with the upper and lower twigs, there are no ants laying pheromone *on their way back* from that particular path. Eventually, on the right, we see several ants are following the path defined by the lower twig, both attracted by, and further multiplying, the steady build-up of pheromone on this path, which is faster than that on the path defined by the upper twig. Although the upper twig provides a fairly short journey, it is more perilous, since this twig is quite narrow, and several ants fall off before being able to strengthen the path.

Actually, this is one of the simplest and most straightforward activities in social insects related to pheromones. The term ‘pheromone’ itself was promoted for this context in 1959 (the late 1950s was clearly a fruitful period for swarm intelligence vocabulary) (Karlson & Luscher, 1959), to encompass the broad range of biologically active chemicals used by insects for varieties of communicative purpose. The context in which we describe it above is known in biology as ‘recruitment’, referring broadly to tasks in which individuals discover an opportunity (usually a food source) and need to recruit others to help exploit it. However there are many other behaviours that are associated with pheromones, such as indicating alarms or warnings, interactions between queens and workers, and mating. An excellent source of further information is Vander Meer et al (1998b), and in particular Vander Meer et al (1998a).

2.1.2 Natural Clustering

Turning now to a different style of swarm behaviour, it is well known that ant species (as well as other insect species) exhibit emergent sorting and clustering of objects. Two of the most well known examples are the clustering of corpses of dead ants into cemeteries (achieved by the species *Lasius Niger*, *Pheidole pallidula*, and *Messor sancta* (Depickere et al, 2004), and the arrangement of larvae into similar-age groups (so called brood-sorting, achieved by *Leptothorax unifasciatus* (Franks & Sendova-Franks, 1992)).

For example, in *Leptothorax unifasciatus* ant colonies, the ants' young are organised into concentric rings called annuli. Each ring comprises young of a different type. The youngest (eggs, and micro-larvae) are clustered together in the central cluster. As we move outwards from the centre, progressively older groups of larvae are encountered. Also, the spaces between these rings increase as we move outwards.

In cemetery formation, certain ant species are known to cluster their dead into piles, with individual piles maintained at least a minimal distance from each other. In this way, corpses are removed from the living surroundings, and cease to be a hindrance to the colony. One aspect of this behaviour in particular is that it is arguably not exemplary of swarm behaviour; i.e. it is perhaps not *collective* intelligence. The explanatory model that seems intuitively correct and confirmed by observation (Theraulaz et al, 2002) is one in which an individual operates

according to quite simple rules while otherwise generally wandering around randomly:

1. If not carrying a corpse, and a single corpse (or quite small cluster of corpses) is encountered, pick it up.
2. If carrying a corpse, and a relatively large cluster of corpses is encountered, put it down.

A single ant could achieve the clustering observed in nature, that seems to operate according to these rules (Theraulaz et al, 2002). However, the emergent clusters are produced faster when a collection of ants are involved. Gaubert et al (2007) is a useful reference for discussion of mathematical and other models of these behaviours. Meanwhile, a steady line of recent research is investigating computational clustering methods that are directly inspired by these natural phenomena (Handl & Meyer, 2007). The first such inspired clustering algorithm was proposed by Lumer & Faieta (1994), closely based on the Deneubourg et al (1991) model of the natural process. In recent work (Handl et al, 2006), an ant-based clustering method called ATTA is tested, and the case is made convincingly that ant-based clustering algorithms certainly have a niche in data mining, performing particularly well on problems where the number of clusters are not known in advance, and where the clusters themselves are highly variable in size and shape. In this article our focus stays with optimisation and we will say little more about clustering; we refer the reader again to Handl and Meyer's recent review (Handl & Meyer, 2007) for further study on this topic.

2.1.3 Natural construction

We now consider the extraordinary collective behaviour that leads to the construction of achievements such as wasp nests, termite mounds and bee hives. Brood-sorting, considered above, exemplifies a simple structure that arises from collective behaviour. However the more visible and impressive structures such as termite mounds have always impressed observers, and often confounded us when we try to imagine how such simple minds can lead to such creations. As will be clear from context, stigmergy seems to be the key to understanding these buildings; patterns inherent in partial elements of structures are thought to trigger simple rule-based behaviour in the insect, which in turn changes the perceived patterns, and so on ..., until a complete hive or nest is built. Much computation-based study has been made of this by Bonabeau, Theraulaz and co-workers.

In nature, the sizes of such social insect structures can reach an astounding 30 metres in diameter (Grassé, 1984). An impressive example of the complexity of these structures comes from the African termites *Macrotermitinae*, 'the fungus

growers'. In a mature nest of this species, there are typically six distinct elements of structure (we adapt this description from Bonabeau et al, 1998):

1. The protective and ribbed cone-shaped outer walls (also featuring ventilation ducts).
2. Brood chambers within the central 'hive' area. They have a laminar structure and contain the nurseries where the young termites are raised.
3. The hive consists of thin horizontal lamellae supported by pillars.
4. A flat floor structure, sometimes exhibiting cooling vents in a spiral formation.
5. The royal chamber: a thick walled enclosure for the queen, with small holes in the walls to allow workers in and out. This is usually well protected underneath the hive structure, and is where the queen lays her eggs.
6. Garden areas dedicated to cultivating fungi. These are arranged around the hive, and have a comb-like structure, arranged between the central hive and the outer walls.
7. Tunnels and galleries constructed both above and below ground which provide pathways from the termite mound to the colony's known foraging sites.

So, how does a collection of termites make such a structure? Perhaps this is the most astonishing example of natural swarm intelligence at work. Observations and descriptions of such structures from the biology literature have tended to focus on *description*, elucidating further and finer detail from a variety of species, but have done little to clarify the mechanism. However, computational simulation work such as Theraulaz & Bonabeau (1995), has indicated how such behaviour can emerge from collections of 'micro-rules', where patterns of the growing structure perceived by an individual termite (or ant, wasp, etc ...) act as a stigmergic trigger, perhaps in tandem with other environmental influences, leading to a specific response that adds a little new structure. Theraulaz and Bonabeau (1995) and Bonabeau et al (2000) have shown how specific collections of such micro-rules can lead to, in simulation, a variety of emergent structures, each of which seems convincingly similar to wasp nests from specific wasp species. Figure 3 (reprinted detail with permission) shows examples of three artificial nests, constructed in this fashion, that closely resemble the nest structures of three specific wasp species;

several more such are presented in Theraulaz & Bonabeau (1995) and Bonabeau et al (2000).

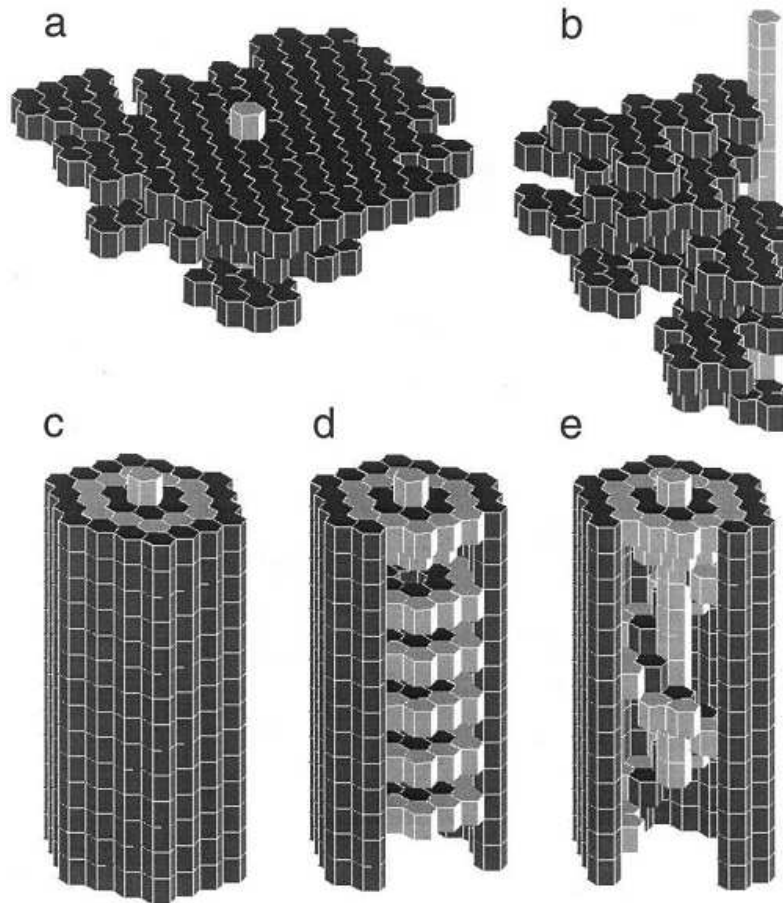


Figure 3. Fig. 2 from Bonabeau et al (2000), reproduced with permission. These show results from artificial colonies of ten wasps, operating under the influence of stimulus-response micro-rules based on patterns in a 3D hexagonal brick lattice. (a) A nest-like architecture resembling the nests of *Vespa* wasps, obtained after 20,000 building steps; (b) An architecture resembling the nests of *Parachartergus* wasps, obtained after 20 000 building steps; (c) resembling *Chartergus* nests, obtained after 100 000 building steps; (d, e) Showing internal structure of (c).

In Figure 4 we see some examples of micro-rules of the kind that can lead to the types of structures shown in Figure 3. A micro-rule simply describes a three-dimensional pattern of ‘bricks’; in the case of the experiments that led to Figure 3, a brick has a hexagonal horizontal cross section, and there are two types of brick (it was observed that at least two different brick types seem necessary for interesting results). The three dimensional pattern describes the immediate neighbourhood of a single central brick, including the seven hexagonal cells above it (the upper patches of hexagons in figure 4) the six that surround it, and the seven below it. In the figure, a ‘white’ hexagonal cell is empty – meaning no brick here; otherwise there are two kinds of bricks, distinguished by different shadings in the figure. A micro-rule expresses the following building instruction: ‘if the substructure defined by this pattern is found, with the central cell empty, than add the indicated type of brick in the central position’.

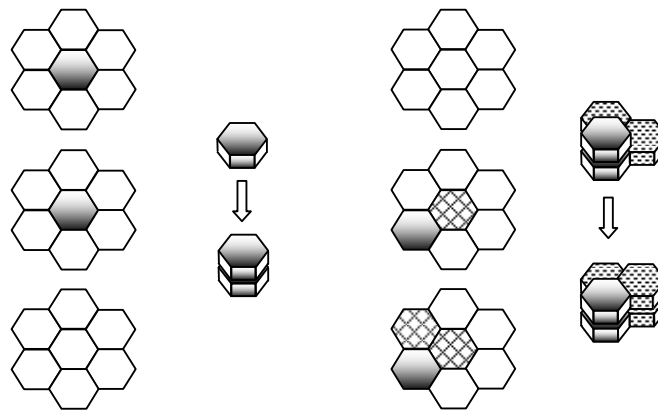


Figure 4. An illustration of two ‘micro-rules’ from the space of such rules that can lead to structures such as those in Figure 3. A single micro-rule defines a building instruction based on matching a pattern of existing bricks. A single column of three groups of 7 hexagonal cells is a micro-rule, by describing a structure around the neighbourhood of the central cell (which is empty in the matching pattern). The building instruction is to fill the central cell with the indicated type of ‘brick’. In this figure, two example micro-rules are shown, each further illustrated by ‘before’ and ‘after’ building patterns on the right.

Intuition suggests how the construction by a collection of agents such as wasps of artefacts such as those in Figure 3, or even more complex artefacts, may be facilitated by specific collections of micro-rules, however that does not make it easy to design a set of micro-rules for a specific target construction. Sets of micro-rules achieving the illustrated results were obtained by using a carefully designed

evolutionary algorithm. Interested readers should consult Bonabeau et al (2000) for further detail, including analyses of the operation of the emerging rulesets, revealing the requirement for various types of co-ordination implicitly built in to the micro-rule collection.

Meanwhile, it is a long way from wasp nests to termite mounds, especially mounds of the complexity hinted at above. However, by considering and extending partial models for elements of termite mounds, in Bonabeau et al (1998) some basis is provided for the suspicion that such complexity may be explained by the interaction of stimulus-response ('micro-rule') based processes and pheromone-based triggers that modify the stimulus-response behaviour, unfolding over time as a controlled morphogenetic process.

Finally we note that much of what we have discussed in this subsection forms part of the basis for the new field of 'swarm robotics'. This area of research (Sahin, 2005; Mondada et al, 2005) focuses on what may be achievable by collections of small, simple robots furnished with relatively non-sophisticated ways to communicate. Chief among the motivating elements of such research are the qualities of robustness, flexibility and scalability that swarm robots could bring to a number of tasks, ranging through agriculture, construction, exploration and other fields. Imagine we wish to build a factory on Mars, for example. We might imagine this could be performed, in some possible future, by a relatively small collection of very sophisticated and intelligent robots. However, in the harsh conditions of Mars, we can expect that loss or destruction of one or more of these is quite likely. Swarms of simple robots, instead, are far more robust to loss and damage, and may present an altogether more manageable, shorter-term and more adaptable approach than the 'super-robot' style.

As yet, swarm robotics has not risen to such heights, although there is published discussion along these lines, drawing from the sources of natural inspiration we have already discussed (Cicirello & Smith, 2001). Meanwhile, interesting and useful behaviours have been demonstrated in swarm robotics projects, after, in almost all cases, considerable work in design, engineering and construction of the individual 'bots'. Supporting these studies, a large part of swarm robotics research is into how to design the individual robots' behaviours in such a way that the swarm (or team) achieves an overall goal or behaviour. Unsurprisingly to many, it turns out that the judicious use of evolutionary computation proves effective for this difficult design problem (Waibell et al, 2009). However, hand-design or alternative principled methods for designing behaviours remains a backbone of this research, especially when the desired overall behaviour is complex, involving many tasks (e.g. Ducatelle et al, 2009), and in general there are several emerging issues in swarm robotics that have sparked active current lines of research, such as the problem of 'interference' – swarm robots, whether co-operating on the same task or not, often physically interfere with each others' operation (Pini et al, 2009) – and the problem of achieving tasks with minimal energy requirements (Roberts et al, 2008).

Relatively impressive behaviours from swarm robotics research has so far included co-operative transport of one or more objects, and co-operation towards moving up a vertical step (as large as the bots involved); readers may visit the European Project 'Swarmbots' and 'Swarmanoids' websites for explanation and many other resources, at <http://www.swarm-bots.org/>, and at <http://www.swarmanoid.org/> as well as similar resources such as <http://www.pherobot.com/>. Hardware and related technology issues remain a bottleneck that still inhibits a full exploration of social insect swarm intelligence in robotics, however this work continues toward that end and will be observed with great interest.

2.2 Foraging

There are broadly two types of natural process that go by the term "foraging", and in turn provide sources of inspiration for optimisation (or resource allocation) methods. In both cases, the overall emergent behaviour is that the swarm finds and exploits good food sources, adaptively moves to good new sources as current ones become depleted, and does all of this with efficient expenditure of energy (as opposed to, for example, brute force search of their environment). The means by which this behaviour is achieved is rather different in these two sources of inspiration.

In one case, that of 'bacterial foraging' (Passino, 2002), individual bacteria are (essentially) directed towards rich areas via *chemotaxis*; that is, they exist in an environment in which their food source diffuses, so they can detect and respond to its presence. In particular, chemotaxis refers to movement along a chemical gradient. An individual e-coli bacterium has helical appendages called *flagellae* which spin either clockwise or anticlockwise (we can think of them as analagous to propellers). When they spin in one direction, the bacterium will 'tumble'; this is an operation which ends up moving the bacterium a short distance, and leaving it with an essentially random new orientation. When the flagellae spin in the other direction, the bacterium's movement will be a 'run' - this is a straight-line movement in the direction the bacterium was facing at the beginning, and continues as long as the flagellae continue to spin in the same direction.

In a nutrient-free and toxin-free environment, an individual bacterium will alternate between clockwise and anti-clockwise movement of its flagellae. So, it tumbles, runs, tumbles, runs, and so forth. The effect of this behaviour is random search for nutrients. However, when the bacterium encounters an increasing nutrient gradient, that is, a higher concentration of nutrient in its direction of movement, its internal chemistry operates in a way that causes the runs to be of longer duration. It still alternates between tumbles and runs, but maintains longer run lengths so long as the gradient continues to increase. The effect of this is to

explore and exploit the food source, moving upwards along the nutrient gradient, while maintaining an element of stochastic exploration.

In addition, under certain conditions we know that bacteria secrete chemicals that attract each other. There is speculation that this can happen in response to nutrient rich environments, so that additional bacteria are recruited to exploit the food source, where the attractive secretions build further on the attraction provided by the chemical gradient. Also, there is evidence that bacteria release such an attractant under stressful conditions too, which in turn may be a protective response; as they swarm into a sizeable cluster, many individuals are protected from the stressful agent. These self-attractant and chemotactic behaviours are known to lead to pattern formation under certain conditions (Budrene & Berg, 1991). These and many other details have been elucidated for *e. coli* and similar bacteria via careful experimentation, for example: Berg & Brown (1972), Segall et al (1986), and deRosier (1998).

The other broad style of efficient collective foraging behaviour is that exhibited by the honeybee (among other insects). When a bee discovers a food source some distance from the hive, it returns to the hive and *communicates the direction, distance and quality* of the food source to other bees. The details of this communication, achieved by specialised ‘dances’, are quite remarkable, and have emerged from a series of ingenious experiments and observations, largely by Karl von Frisch (1967). The essential details are these: in context, the dance is performed in alignment with a particular aspect of the hive structure, which provides an absolute reference against which the bee audience can perceive specific angles. The main dance is the ‘waggle’ dance, which consists of a straight line movement, during which the bee waggles from side to side along the way. This straight line movement is done upwards at a particular angle from the vertical. At the end of the straight line part, the bee loops round to the starting point and repeats (actually, it alternates the direction of this loop, drawing a figure "8"). The angle of this dance from vertical indicates to the bee audience the direction they need to take with respect to the current position of the Sun. Among various extraordinary aspects of this, it is known that the bee automatically corrects for movement of the Sun during the day, and communicates the correct direction. Also, at times, the bee will pause its dance, and allow watching bees to sample the nectar it is carrying, giving an indication of the quality of the food source.

More interesting from the algorithmic viewpoint, however, is that the abundance of the food source is communicated by the duration of the dance (essentially, the number of times the figure "8" is repeated). An individual enjoying this performance may or may not decide to follow these directions and be ‘recruited’ to this particular source. Such an individual may also be exposed to rival performances. However, the longer the duration, the more bees will see this dance, and the more will be recruited to this dance rather than others. In this way, the bee colony sports the emergent behaviour of smart resource allocation, with

more bees assigned to better sources, and adaptation over time as returning bees gradually provide shorter and shorter dances as the source becomes depleted.

As we will see later, both bee and bacterial foraging have been taken as the inspiration for general optimisation methods, as well as for approaches to the specific problem domain of optimal resource allocation.

2.3 Flocking

Perhaps the most visible phenomenon that brings to mind swarm intelligence is the travelling behaviour of groups (flocks, swarms, herds, etc...) of individuals that we are all familiar with. The mesmerising behaviour of large flocks of starlings is a common morning sight over river estuaries. Swarms of billions of monarch butterflies, herds of wildebeeste, schools of tunafish, swarms of bees, all share common emergent behaviours, chiefly being:

1. the individuals stay close to each other, but not too close, and there seem to be no collisions;
2. swarms change direction smoothly, as if the swarm was a single organism;
3. *unlike* a single organism, yet still smoothly and cleanly, swarms sometimes pass directly through narrow obstacles (in the way that a stream of water passes around a vertical stick placed centrally in the stream's path).

In some ways, such swarm behaviour is arguably less mysterious than other emergent behaviours; it seems clear that we might be able to explain this behaviour via a built-in predisposition for individuals to stay with their colleagues, and we can readily imagine how evolution will have favoured such behaviour: there is safety in numbers. However, the devil is in the detail, and it took seminal work by Reynolds (1987) to outline and demonstrate convincing mechanisms that can explain these behaviours. Reynolds' work was within the computer graphics community, and has had a volcanic impact there. Now known as 'Reynold's rules', the recipe that achieves realistic swarm behaviour (with some, but not obtrusively much, parameter investigation needed depending on the species simulated) is this triplet of steering behaviours to be followed by each individual in a swarm:

Separation: steer to avoid coming too close to others.

Alignment: steer towards the mean heading of others.

Cohesion: steer towards the mean *position* of others

A basic illustration of each rule is given in Figure 4. In the figure, we take the common terminology of 'boid' to refer to an individual in a flock. The figure shows examples of the adjustments that might be made under the guidance of the

rules. To understand how realistic swarm simulation works, it is important to note that each boid has its own perceptual field – i.e. it could only ‘see’ a certain distance, and had a specific field of view (boids cannot see behind them, for example). The adjustments it makes to its velocity at any time are therefore a function of the positions and velocities of the boids in its perceptual field, rather than a function of the flock as a whole.

The rules are key ingredients to a realistic appearance in simulated flocks, but there are several other details, particularly regarding obstacle avoidance and goal-seeking behaviour. Interested readers may consult Reynolds (1987) and the many papers that cite it. It is important to note that these rules are not strictly nature-inspired, in the sense that Reynolds was not attempting to explain natural swarming behaviour, he was simply attempting to emulate it. However, the resulting behaviour was found to agree well with observations of natural flocking behaviour (e.g. Partidge (1982) and Potts (1984)), and Reynolds (1987) reported that “many people who view these animated flocks immediately recognize them as a representation of a natural flock, and find them similarly delightful to watch”. These techniques are now common in the film industry; among the earliest uses were in the film *Batman Returns* (1992, director Tim Burton), in which Reynolds’ rules lay behind the simulated bat swarms and flocks of penguins.

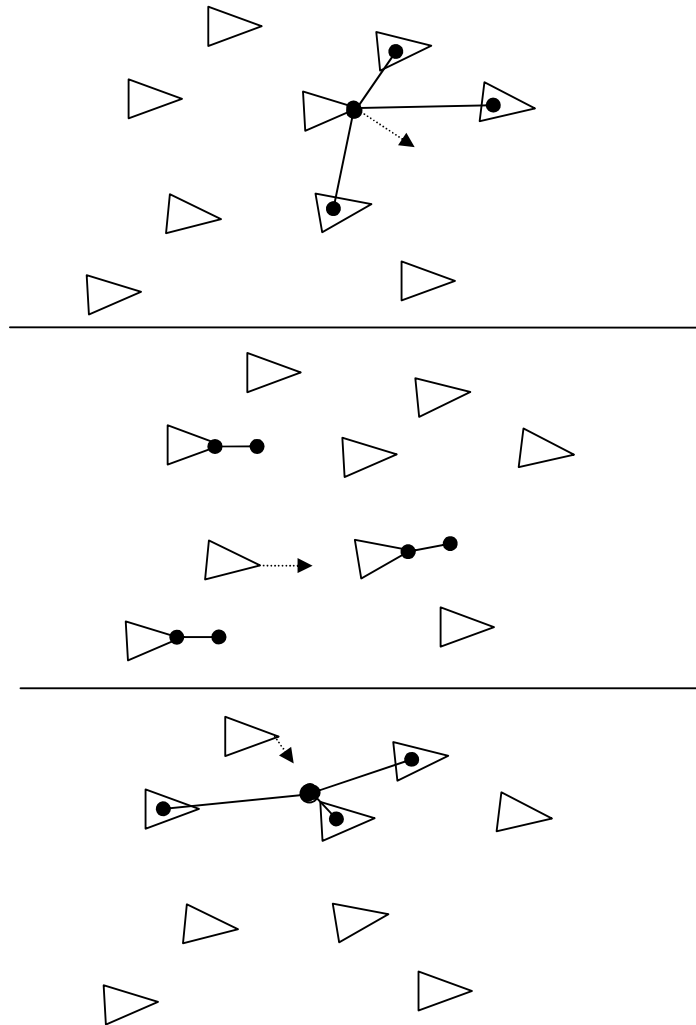


Figure 5. Illustrating Reynolds' rules, which lead to natural-looking behaviour in simulated swarms. Upper: Separation: each boid makes an adjustment to velocity which prevents it coming too close to the flockmates in its perceptual field. Middle: Alignment: a boid adjusts its heading towards the average of those in its perceptual field; lower: a boid makes an adjustment to velocity that moves it towards the mean position of the flockmates in its perceptual field.

Meanwhile, natural flocking behaviour also turns out to be one of the sources of inspiration for the highly popular and successful particle swarm optimisation

algorithm, which appears in the next section as one of the prominent flagships for swarm intelligence. It is not obvious why flocking behaviour might lead to an optimisation algorithm, however it soon becomes clear when we consider the dynamics of flocking, and the tendency of optimisation landscapes to be locally smooth. In the case of bacterial foraging, the dynamics of the natural behaviour are such that individuals will tend to congregate around good areas. With the bacterial example, nature provides mechanisms for suggesting appropriate directions of movement, while there is a clear goal for the bacterial colony to achieve - find nutrient rich (and toxin free) areas. In particular for the current context, the secretion of attractant chemicals is a mechanism that promotes bacteria swarming together, while an individual's position in its environment directly provides it with a level of 'fitness' that it can sense in terms of nutrient concentration.

When we consider flocking behaviour in birds, however, Reynolds' work provides clues about appropriate ways to move together as a swarm, but there is no clear mirror of a 'fitness' in the environment. Often birds will migrate from A to B, knowing where they are going, rather than seeking new environments. However, if flocks *did* have a goal to move towards 'fitter' positions in the landscape they travel, then it becomes intuitively reasonable to consider the cohesive swarm behaviour as a sensible way to achieve local exploration around fit areas, perhaps enabling the sensing of even fitter areas that may then sway the overall movement of the flock. In this way, flocking behaviour combines with a little algorithm engineering to achieve a very successful optimisation mechanism.

3. Two Main Concepts for Swarm Intelligence Algorithms

When we consider the impact of swarm intelligence so far on computer science, two families of algorithms clearly stand out in terms of the amount of work published, degree of current activity, and the overall impact on industry. One such family is inspired directly by the pheromone-trail following behaviour of ant species, and this field is known as Ant Colony Optimization (ACO). The other such family is inspired by flocking and swarming behaviour, and the main exemplar algorithm family is known as Particle Swarm Optimization (PSO). Also in this family are algorithms based on bacterial foraging, and some of the algorithms that are based on bee foraging; these share with PSO the broad way in which the natural phenomenon is mapped onto the concept of search within a landscape. In this section we discuss these two main families in turn.

3.1 Ant Colony Optimization

Ant Colony Optimization (ACO) was introduced in 1996 via an algorithm called 'Ant System' (AS) (Dorigo et al, 1996). The basic approach used in AS remains highly characteristic of most ACO methods in current use, and we will describe it next.

Recall that, in the natural case, an ant finds a path from its nest to a food source by following the influences of pheromone trails laid down by previous ants who have previously sought food (and usually returned). AS, and ACO algorithms in general, mirror aspects of this behaviour quite faithfully. In short: an artificial ant builds a solution to the optimisation problem at hand, and lays down simple 'artificial pheromone' along the route it took towards that solution. Following artificial ants then build solutions of their own, but are influenced by the pheromone trails left behind by their precursors. This is the essential idea, and starts to indicate the mapping from the natural to the artificial case. However, there are various further issues necessary to consider to make this an effective optimization algorithm. We discuss this further in the next section, focussing on the mapping from the basic ideas of ACO to applications in optimisation.

3.1.1 Applying ACO to optimisation problems

In order to apply AS to an optimisation problem, the problem needs to be represented in such a way that any candidate solution to it is a specific path along a network. This network can be conceived as having a single start node, from which (usually) every ant starts, and a single finish node, reaching which indicates that the path taken encodes a complete solution. A clear example might be the network of roads in a city, where each junction of roads is a node in the network, and each road is an arc or edge between nodes. Consider the problem of finding the shortest path between a specific junction A and another specific junction B. In this case, A and B are clearly the start and finish nodes, and we can imagine an ACO approach which maps very closely indeed to the natural case. However, with a little thought, it is clear that we should constrain each individual ants' path construction so that it does not return to a junction it has already visited (unless this is a valid move for the problem under consideration). Also, though we might choose to simulate the preferential recruitment of new ants to shorter paths by closely following the natural case, it seems more sensible and straightforward to make the pheromone trail strength directly a function of the solution quality. That is, when an ant has completed its path, we evaluate the quality of its solution, and render things such that better solutions lead to stronger pheromone deposits along its arcs. These pheromone deposits will decay over time, however, just as in the natural case - we can see that this will prevent premature convergence to poor solutions that happen to be popular in the early stages.

Finally, since such information is often available to us, and would seem useful in cases where ants have large numbers of choices, we might bias the paths available at each junction with the aid of a simple heuristic evaluation of the potential of that arc. For a shortest path problem, for example, this could be based on how much closer to B each arc would leave the ant.

With such considerations in mind, we can envisage artificial ants travelling the road network from A to B via distinct but sensible routes. At each junction, the ant senses the pheromone levels that await it at each of the arcs it can feasibly take. These levels are made from many components; arcs that are highly attractive will probably enjoy the remnants of trails from prior ants that have reported good

solutions, and/or may have a good heuristic component. Arcs with low pheromone levels will probably be losers in the heuristic stakes, and have seen little activity that has led to good solutions; however, the ant may still choose such an arc, since our algorithm is stochastic.

Finally, as an individual artificial ant arrives at B, it retrospectively lays pheromone on the path it took, where the strength of that pheromone trail will reflect the quality of its solution. The next artificial ant starts from A and sees a slightly updated pheromone trail (stigmergic) pattern, and so it continues.

To apply this method to other problems, we simply need (implicitly, at least) a network-based representation of the problem as described. If we are solving the traveling salesperson problem (TSP), for example, the network is the complete graph of the cities, each arc between cities indicates the distance or cost of that arc, and in this case an individual ant can start anywhere. As we follow an individual ant's route, we sensibly constrain its potential next-hops to avoid cycles, and along the way we may bias its choices simply by using the distance of the next arc, and it retrospectively lays pheromone once it has completed a tour of all cities. In general, an optimisation problem can always be approached in this way, by suitable choices of semantics for nodes and arcs, and well designed routines for generating and constraining an ant's available choices at each junction.

We can now finish this explication by clarifying the AS algorithm, which in fact has already been covered verbosely in the above. Once the transformation of the problem has been designed, so that an ant's path through a network provides a candidate solution, the algorithm cycles through the following two steps until a suitable termination condition is reached:

Solution Construction: a number of ants individually construct solutions based on the current pheromone trail strengths (initially, pheromone is randomly distributed). Each ant steps through the network choosing among feasible paths. At each choice point, the ant chooses among available arcs according to a function of the pheromone strength on each arc, and of the heuristic values of each arc. In the original, and still commonly used version of this function (see Dorigo et al (1996) and many more), the pheromone and heuristic components for each arc are exponentiated to parameters α and β respectively, allowing for tuning of the level to which the algorithm relies on exploration and heuristic support. Also, the overall attractiveness value of each arc is scaled so that the ant can treat these values as a probability distribution over its available choices.

Pheromone Update: When the ants' paths are complete, for each ant, the corresponding solution is evaluated, and pheromone is laid on each arc travelled in proportion to the overall solution quality. Also, the component of pheromone strength that arises from earlier ants is reduced. Quite simply, for any particular arc, its updated pheromone strength p_{new} is $(1-\rho)p_{\text{old}} + \rho f$, where ρ controls the speed of pheromone decay, and f

accumulates the overall quality of solutions found which involved that arc in the current iteration.

ACO has now been applied to very many problems, and (clearly, or we would probably not have devoted so much time to it) has been very successful, especially when hybridized with local search or some other meta-heuristic (in such hybridized algorithms, an ant will typically use the additional heuristic for a short time to find an improvement locally to its solution). Initially demonstrated for the TSP (Dorigo et al, 1996), there are an enormous number of applications of ACO now published. We mention vehicle routing (Gambardella et al, 1999), rule discovery (Parpinelli et al, 2002), and protein/ligand docking (Korb et al, 2007), just to give some initial idea of the range of applications. To discover more, recent surveys include Blum (2005) and Gutjahr (2007), the latter concentrating on theoretical analyses. Meanwhile, Socha & Dorigo (2008) show how to apply ACO to continuous domains (essentially, ants select parameters via a probability density function, rather than a discrete distribution over a fixed set of arcs).

3.1.2 Ant-Based Routing in Telecommunications

The basic ACO idea of exploiting pheromone-trail based recruitment is also the inspiration for a healthy sub-area of research in communications networks; therein, ant-inspired algorithms are designed to assist with network routing and other network tasks, leading to systems that combine high performance with a high level of robustness, able to adapt with current network traffic and robust to network damage. Early and prominent studies in this line were by Schoonderwoerd et al (1996; 1997), which were soon built upon by di Caro and Dorigo's AntNet (1998). To explain this application area, and the way that ACO ideas are applied therein, it will be helpful to first explore the problem and the associated solution that was studied in Schoonderwoerd et al's seminal work.

Schoonderwoerd et al were concerned with load balancing in telecommunication networks. The task of a telecommunication network is to connect calls that can arise at any node, and which may need to be routed to any other node. The networks themselves, as a function of the capacities of the constituent equipment at each node, cannot guarantee successful call connections in all cases, but they aim to maintain overall acceptable performance under standard conditions. At very busy times, and/or if a particular node is overwhelmingly flooded with calls, then typically many calls will be 'dropped'. It is worth noting that there are problems with central control of such systems (landline telecommunication networks, mobile networks, traffic networks, and so forth ...). To achieve centralised control in a way that manages load-balancing or any other such target, several disadvantages are apparent. The controller usually needs current knowledge about many aspects of the entire system, which in turn necessitates using communication links from every part of the system to the controller itself. Centrally managed control mechanisms therefore scale badly as a result of the rapidly increasing overheads of this communication, interfering with

the performance of the system itself. Also, failure in the controller will often lead to complete failure of the complete system.

Schoonderwoerd et al's ant-inspired approach, which remains a central part of the majority of more recent ant-based approaches, was to replace the routing tables in such networks with so-called 'pheromone tables'. Networks of the type of interest invariably have a routing table at each node, specifying which 'next-hop' neighbouring node to pass an incoming call to, given the ultimate destination of that call. In Schoonderwoerd et al's 'Ant-Based Control' (ABC) method, the routing table at a network node instead provided n probability distributions over its neighbouring nodes, one for each of the n possible destinations in the network. When a routing decision is to be made, it is made stochastically according to these probabilities – i.e. it is most likely that the next-hop with the highest probability will be taken, but there is a chance that the next-hop with the lowest probability will be taken instead. The entries in the pheromone table were considered analogous to pheromone trail strengths, and changed adaptively during the operation of the network. Updating of these trails in ABC is very simple – whenever a call is routed from node A to B , the entries for A in node B 's routing table are all increased, with corresponding reductions to other entries. However this simple idea has obvious intuitive benefits; first, by testing various routing decisions over time (rather than deterministic decisions), the process effectively monitors the current health of a wide variety of different routing strategies; when a link is over-used, or down, this naturally leads to diminution in its probability of use, since the associated entries in routing tables will not be updated, and hence will naturally reduce as alternatives are updated. Also, as it turns out, the pheromone levels can adapt quite quickly to changes in call patterns and loads. The ABC strategy turned out to be surprisingly effective, despite its simplicity, when Schoonderwoerd et al compared it with a contemporary agent-based strategy developed by Appleby and Steward (1994), and found it superior over a wide range of different situations.

Research in ant-based approaches for decentralised management is increasingly very active (e.g. Hossein and Saadawi, 2003; Rosati et al, 2008; Di Caro et al, 2008). The essential idea, to replace static built-in routing strategies with stochastic 'pheromone tables' or similar, is applicable in almost all modern communication scenarios, ranging through ad-hoc computer networks, mobile telephone networks, and various layers of the internet. Ongoing research continues to explore alternative strategies for making the routing decisions, controlling the updates to pheromone trails, and so forth, while investigating various distinct application domains, and continuing to find competitive or better performance than alternative state of the art methods used in network engineering.

3.2 Particle Swarm Optimization and Foraging

Particle swarm optimization (PSO) was established in 1995 with Kennedy and Eberhart's paper in IJCNN (Kennedy & Eberhart, 1995). The paper described a rather simple algorithm (and time has seen no need to alter its straightforward fundamentals), citing Craig Reynolds' work as inspiration (Reynolds, 1987), along with slightly later work in the modelling of bird flocks (Heppner & Grenander, 1990). The basic idea is to unite the following two notions: (i) the behaviour of a flock of birds moving in 3D space towards some goal; (ii) a swarm of solutions to an optimisation problem, moving through the multidimensional search space towards good solutions.

Thus, we equate a 'particle' with a candidate solution to an optimization problem. Such a particle has both a *position* and a *velocity*. Its position is, in fact, precisely the candidate solution it currently represents. Its velocity is a displacement vector in the search space, which (it hopes) will contribute towards a fruitful change in its position at the next iteration.

The heart of the classic PSO algorithm is in the step which calculates a new position for the particle based on three influences. The inspiration from Reynolds (1987) is clear, but the details are quite different, and, of course, exploit the fact that the particle is moving in a search space and can measure the 'fitness' of any position. The influences - the components that lead to the updated position - are:

Current velocity: the particle's current velocity (obviously);

Personal Best: the particle remembers the fittest position it has yet encountered, called the personal best. A component of its updated velocity is the direction from its current position to the personal best;

Global Best: every particle in the swarm is aware of the best position that any particle has yet discovered (i.e. the best of the personal bests). The final component of velocity update, shared by all particles, is a vector in the direction from its current position to this globally best known position.

Following a random initialisation of positions and velocities, evaluation of the fitness of the particles' current positions, and consequent initialisation of the personal bests and global best, PSO proceeds in a remarkably straightforward manner. First, each particle updates its velocity by adding a vector in each of the above three component directions. To provide these vectors, in the classic algorithm, the current velocity component is left undisturbed, while the personal and global best components are each scaled by a random scalar drawn uniformly from [0,2]. The resulting vector is used to update the current velocity, and the new velocity vector is used to update the current position. The new position is evaluated, bookkeeping is done to update personal and global bests, and then we repeat.

Kennedy and Eberhart initially reported that PSO appeared to do very well over a wide range of test problems, including its use as an alternative to backpropagation for training an artificial neural network (Kennedy & Eberhart, 1995). Perhaps helped by the ease of implementation of this algorithm (remarkably few lines of code are needed for the classic algorithm), an avalanche of papers began to follow, almost invariably adding to the evidence that this algorithm provides a very substantial contribution to optimization practice. Naturally, this field is now rich in variants and extensions to the original design -- a number of recent surveys are available (e.g. Reyes-Sierra & Coello, 2006; Yang et al, 2007) -- while the published applications are as varied as one might expect from such a generally applicable algorithm.

3.2.1 Bacteria and Bees

Newer to the ranks of swarm-intelligence based optimisation, and yet to prove quite as widely successful, are techniques inspired by bacterial and bee foraging. For the most part, these algorithms follow the broad direction of PSO, in that individuals in a swarm represent solutions moving through a landscape, with the fitness of their current solution easily obtained by evaluating their position. Meanwhile, just as with PSO, an individual's movement through this landscape is influenced by the movements and discoveries of other individuals. The fine details of a Bacterial Foraging Algorithm (BFA), however, are quite distinct, and in one of the more popular methods draw quite closely from what is known (and briefly touched upon above) about bacterial swarming in nature. Passino (2002) presents a fine and detailed explication of both the natural case and the BFA. It turns out that BFA-style algorithms are enjoying quite some success in recent application to a range of engineering problems (e.g. Niu et al, 2006; Tripathy & Mishra, 2007; Guney & Basbug, 2008).

Also inspiring, so far, a small following are algorithms that are inspired by bee foraging behaviour. The authoritative sources for this are Quijano and Passino's papers respectively outlining the design and theory (Quijano & Passino, 2007a) and application (Quijano & Passino, 2007b) of a bee foraging algorithm. In Quijano & Passino (2007a) the design of a bee foraging algorithm is presented in intimate connection with an elaboration of the mechanisms of natural bee foraging (such as we briefly described earlier). The algorithm is as much a model of the natural process as it is a routine applicable to certain kinds of problem. Considering individual bees as resources, the concept here is to use bee foraging behaviour as a way to ideally distribute those resources in the environment, and maintain an ideal distribution over time as it adapts to changing patterns of supply. Just as natural bees maintain an efficient distribution of individuals among the available sources of nectar, the idea is that this can be mapped to control problems which aim to maintain a distribution of resources (such as power or voltage) in such a way that some goal is maintained (such as even temperature or maximal efficiency). In Quijano & Passino, (2007b), we see the algorithm tested successfully on a dynamic voltage allocation problem, in which the task is to

maintain a uniform and maximal temperature across an interconnected grid of temperature zones.

Finally, we note that bee foraging behaviour has also directly inspired techniques for internet search, again, based on the notion of maintaining a maximally effective use of server resources, adapting appropriately and effectively to the relative richness of new discoveries (Walker, 2000; Nakrani & Tovey, 2003).

4. Current Trends and Concluding Notes

We have pointed to a number of survey papers and other works from which the reader can attain a full grasp of the current activity in swarm intelligence algorithms, but in this brief section we attempt a few notes that outline major current trends, and then we wrap up.

A notable trend in recent work on particle swarm optimisation, and indeed on metaheuristics in general, is towards the creation of hybrid algorithms. While themes from evolutionary computation continue to be incorporated in PSO (Shi et al, 2005), others have explored the idea of hybridization with less frequently used techniques such as scatter search and path relinking (Yin et al, 2007), immune system methods (Zhang & Wu, 2008) and, indeed, ant colony optimization (Holden & Freitas, 2007). Meanwhile, the range of problems to which PSO may be applied has been greatly increased with the development of multi-objective forms of PSO (Coello et al, 2004).

Other work has involved the use of multiple swarms. This may allow each swarm to optimize a different section of the solution (van den Bergh & Engelbrecht, 2004). Alternatively, each swarm may be configured differently to take advantage of the strengths of different PSO variants (e.g. Jordan et al, 2008), in an attempt to create a more reliable algorithm that can be applied to a wide range of problem domains.

The themes of multi-objective optimization and hybridization equally apply to recent research into ant-colony optimisation. While multi-objective ACO is a more mature field than multi-objective PSO (see, for example, MARIANO & Morales, 1999), work continues in categorizing and comparing multi-objective approaches to ACO (e.g. Garcia-Martinez et al, 2007), in creating generic frameworks for multi-objective ACO and in creating new multi-objective variants (e.g. Alaya, 2007). Recent applications have seen single objective ACO hybridized with genetic algorithms (Lee et al, 2008), beam search (Blum, 2005b), and immune systems (Yuan et al, 2008) and multi-objective ACO used in combination with dynamic programming (Häckel et al, 2008) and integer linear programming (Doerner et al, 2006).

Other recent work has seen ACO adapted for use in continuous domains (Dreo Siarry, 2006; Socha & Dorigo, 2008), while research continues into variations of ACO and new algorithm features, for example, different types of pheromone and the use of dominance rules to warn ants from searching amongst solutions known to be of low quality (Lin et al, 2008).

Recent work on bacterial foraging algorithms has concentrated on exploiting the effectiveness of the local search ability of the algorithm, while adapting it improve the global search ability on high dimensional and multi-modal problems. With this aim, bacterial foraging has been hybridized with more effective global optimizers such as genetic algorithms (Chen et al, 2007; Kim et al, 2007) and particle swarm optimization (Tang et al, 2007; Biswas et al, 2007).

In conclusion, we have attempted to demystify the concept of swarm intelligence, and, after touring through the chief sources of natural inspiration, distilled the essence of its impact and presence in computer science down to two major families of algorithms for optimisation. No less intriguing and exciting additional topics in the swarm intelligence arena, that we have also discussed, are stigmergic construction, ant-based clustering, and swarm robotics. It is abundantly clear that the natural inspirations from swarming ants, bees and birds (among others) have provided us with new ideas for optimisation algorithms that have extended the state of the art in performance on many problems, sometimes with and sometimes without additional tailoring and hybridization. Ant-based clustering seems also to provide a valuable contribution, while swarm robotics, stigmergy based construction, and a variety of other emerging subtopics have considerable promise, and will doubtless develop in directions rather difficult to foresee.

References

- Inés Alaya (2007) Ant Colony Optimization for Multi-objective Optimization Problems, in Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, pages 450–457, 2007.
- Appleby, S., & Steward, S. (1994). Mobile software agents for control in telecommunications Networks. *BT Technology Journal*, Vol. 12, No.2.
- H. Berg and D. Brown (1972) Chemotaxis in *Escherichia coli* analysed by three-dimensional tracking. *Nature*, **239**, pp. 500–504, 1972.
- Arijit Biswas, Sambarta Dasgupta, Swagatam Das, and Ajith Abraham. (2007), Synergy of PSO and Bacterial Foraging Optimization — A Comparative Study on Numerical Benchmarks, in E. Corchado et al., ed., *Innovations in Hybrid Intelligent Systems*, no. 44 in *Advances in Soft computing*, pp. 255–263, 2007.
- Christian Blum (2005a) Ant colony optimization: Introduction and recent trends, *Physics of Life Reviews*, **2**(4), pp. 353–373, 2005
- Christian Blum (2005b) Beam-ACO---hybridizing ant colony optimization with beam search: an application to open shop scheduling, *Computers & Operations Research*, pages 1565–1591, 2005.
- Eric Bonabeau, Sylvain Guérin, Dominique Snyers, Pascale Kuntz and Guy Theraulaz. (2000) Three-dimensional architectures grown by simple ‘stigmergic’ agents, *Biosystems*, **56**, pp. 13–32, 2000.
- Eric Bonabeau, Guy Theraulaz, Jean-Louis Deneubourg, Serge Aron, and Scott Camazine (1997), Self-organization in social insects, *Trends in Ecology and Evolution*, **12**(5), pp. 188–193, 1997.
- Eric Bonabeau, Guy Theraulaz, Jean-Louis Deneubourg, Nigel R. Franks, Oliver Rafelsberger, Jean-Louis Joly and Stephane Blanco (1998) A model for the

- emergence of pillars, walls and royal chambers in termite nests *Philosophical Transactions of the Royal Society B: Biological Sciences*, **353**(1375), pp. 1561—1576, 1998.
- E. Budrene and H. Berg (1991) Complex patterns formed by motile cells of *escherichia coli*. *Nature*, **349**, pp. 630—633, 1991.
- Tai-Chen Chen, Pei-Wei Tsai, Shu-Chuan Chu, and Jeng-Shyang Pan (2007), *A Novel Optimization Approach: Bacterial-GA Foraging*, in *Proceedings of the Second International Conference on Innovative Computing, Information and Control (ICICIC)*, page 391. IEEE Computer Press, 2007.
- Vincent A. Cicirello and Stephen F. Smith (2001) Wasp nests for self-configurable factories, in *Proc. 5th Int. Conf. on Autonomous Agents*, pp. 473—480, ACM, 2001.
- Carlos A.-Coello Coello, Gregorio-Toscano Pulido, and Maximino-Salazar Lechuga (2004) Handling Multiple Objectives With Particle Swarm Optimization, *IEEE Transactions on Evolutionary Computation*, **8**(3):256--279, 2004.
- Coloni A., M. Dorigo & V. Maniezzo (1992a). Distributed Optimization by Ant Colonies, *Proceedings of the First European Conference on Artificial Life*, Paris, France, F.Varela and P.Bourgine (Eds.), Elsevier Publishing, 134-142.
- Coloni A., M. Dorigo & V. Maniezzo (1992b). An Investigation of some Properties of an Ant Algorithm. *Proceedings of the Parallel Problem Solving from Nature Conference (PPSN 92)*, Brussels, Belgium, R.Männer and B.Manderick (Eds.), Elsevier Publishing, 509-520.
- J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain and L. Chretien. (1991) The dynamics of collective sorting: robot-like ants and ant-like robots, in *From Animals to Animats: Proc. 1st Int. Conf. on Simulation of Adaptive Behaviour*, Jean Arcady-Meyer and Stewart Wilson (eds.), MIT Press, Cambridge, pp. 356—365, 1991.
- S. Depickere, D. Fresneau and J.-L. Deneubourg (2004) Dynamics of aggregation in *Lasius niger* (Formicidae): influence of polyethism, *Insect. Sociaux*, **51**(1), 81–90, 2004.
- D. DeRosier (1998) The turn of the screw: The bacterial flagellar motor, *Cell*, **93**, pp. 17—20, 1998.
- Gianni Di Caro and Marco Dorigo (1998) AntNet: Distributed Stigmergetic Control for Communications Networks. *JAIR*, **9**, pp. 317—365, 1998.
- Giann Di Caro, Frederick Ducatelle and Luca M. Gambardella, Theory and practice of Ant Colony Optimization for routing in dynamic telecommunications networks, in Sala N., Orsucci F. (Eds.), *Reflecting Interfaces: the Complex Coevolution of Information Technology Ecosystems*, Idea Group, Hershey, PA, USA, 2008
- K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C.Strauss, and C.Stummer (2006), Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection, *European Journal of Operational Research*, **171**:830--841, 2006.
- Dorigo M., V. Maniezzo & A. Coloni (1991). The Ant System: An Autocatalytic Optimizing Process. *Technical Report No. 91-016 Revised*, Politecnico di Milano, Italy
- Marco Dorigo, V. Maniezzo and A. Coloni (1996) Ant system: optimization by a colony of co-operating agents. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics*, **26**(1), pp. 29—41, 1996.
- J. Dreio and P. Siarry, An ant colony algorithm aimed at dynamic continuous optimization, *Applied Mathematics and Computation*, **181**:457--467, 2006.
- Ducatelle F., Förster A., Di Caro G., Gambardella L.M. New task allocation methods for robotic swarms, In *9th IEEE/RAS Conference on Autonomous Robot Systems and Competitions*, May, 2009

- N.R. Franks and A. Sendova-Franks, Brood sorting by ants: distributing the workload over the work-surface. *Behav. Ecol. Sociobiol.*, **30**(2), 109–123, 1992.
- Karl von Frisch, *The Dance Language and Orientation of Bees*, Harvard University Press, Cambridge, MA, 1967.
- Luca Maria Gambardella, Éric Taillard and Giovanni Agazzi, MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows, in *New Ideas in Optimization*, D. Corne, M. Dorigo, F. Glover (eds.), McGraw Hill, London, pp. 63–76, 1999.
- C. García-Martínez, O. Cordon, and F. Herrera, A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP, *European Journal of Operational Research*, **180**:116–148, 2007.
- Laurent Gaubert, Pascal Redou, Fabrice Harrouet and Jacques Tisseau, A first mathematical model of brood sorting by ants: Functional self organisation without swarm-intelligence. *Ecological Complexity*, **4**, pp. 234–241 2007.
- Pierre-Paul Grassé, La reconstruction du nid et les coordinations inter-individuelles chez *Bellicositermes Natalensis* et *Cubitermes* sp. La théorie de la stigmergie: essai d'interprétation du comportement des termites constructeurs, *Insect Soc.*, **6**, pp. 41–84, 1959.
- Pierre-Paul Grassé, *Termitologia, Tome II -- Fondation des sociétés construction*, Paris: Masson, 1984.
- K. Guney and S. Basbug, Interference suppression of linear antenna arrays by amplitude-only control using a bacterial foraging algorithm, *Progress In Electromagnetics Research*, **79**, pp. 475–497, 2008.
- Walter J. Gutjahr, Mathematical runtime analysis of ACO algorithms: survey on an emerging issue, *Swarm Intelligence*, **1**(1), pp. 59–79, 2007.
- Sascha Häckel, Marco Fischer, David Zechel, and Tobias Teich, A Multi-Objective Ant Colony Approach for Pareto-Optimization Using Dynamic Programming, in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 33–40. ACM, 2008.
- Julia Handl and Bernd Meyer, Ant-based and swarm-based clustering, *Swarm Intelligence*, **1**(2), pp. 95–113, 2007.
- Julia Handl, Joshua Knowles and Marco Dorigo, Ant-based clustering and topographic mapping. *Artificial Life*, **12**(1), pp. 35–61, 2006.
- F. Heppner and U. Grenander, A stochastic nonlinear model for coordinated bird flocks, in *The Ubiquity of Chaos*, S. Krasner (ed.), AAAS Publications, Washington, 1990.
- Nicholas Holden and Alex A. Frietas, A Hybrid PSO/ACO Algorithm for Classification, in *Proceedings of the 2007 GECCO Conference Companion on Genetic and Evolutionary Computation*, pages 2745–2750, 2007.
- O. Hussein and T. Saadawi., Ant routing algorithm for mobile ad-hoc networks (ARAMA). In *Proc. IEEE Conf. on Performance, Computing and Communications*, pp. 281–290, 2003.
- Johannes Jordan, Sabine Helwig, and Rolf Wanka, Social Interaction in Particle Swarm Optimization, the Ranked FIPS, and Adaptive Multi-Swarms, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 49–56, 2008.
- P. Karlson and M. Luscher, Pheromones: a new term for a class of biologically active substances, *Nature*, **183**, pp. 155–176, 1959..
- James Kennedy and Russ Eberhart, Particle swarm optimization, in *Proc. IEEE International Joint Conference on Neural Networks*, IEEE Press, Piscataway, pp. 1942–1948, 1995.

- Dong-Hwa Kim, Ajith Abraham, and Jae-Hoon Cho, A hybrid genetic algorithm and bacterial foraging approach for global optimization, *Information Sciences*, **177**:3918—3937, 2007.
- Oliver Korb, Thomas Stützle and Thomas E. Exner, An ant colony optimization approach to flexible protein-ligand docking, *Swarm Intelligence*, **1**(2), pp. 115—134, 2007.
- Zne-Jung Lee, Shun-Feng Su, Chen-Chia Chuang, and Kuan-Hung Liu, Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment, *Applied Soft Computing*, **8**:55--78, 2008.
- B.M.T. Lin, C.Y. Lu, S.J. Shyu, and C.Y. Tsai, Development of new features of ant colony optimization for flowshop scheduling, *International Journal of Production Economics*, **112**:742—755, 2008.
- E. Lumer and B. Faieta, Diversity and adaptation in populations of clustering ants, in *From Animals to Animats 3: Proc. of 3rd Int. Conf. on Simulation of Adaptive Behaviour*, D. Cliff et al (eds.), MIT Press, pp. 501—508, 1994.
- C.E. Mariano and E.~Morales, MOAQ: An Ant-Q algorithm for multiple objective optimization problems, in W. Banzhaf, J. Daida, A. E. Eiben, M.H. Garzon, V. Hnavor, M.~Jakiela, and R.E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computing Conference (GECCO 99)*, pages 894—901, 1999.
- Francesco Mondada, Luca Maria Gambardella, Dario Floreano, Stefano Nolfi, Jean-Louis Deneubourg and Marco Dorigo, The Cooperation of Swarm-Bots: Physical Interactions in Collective Robotics, in *IEEE Robotics and Automation Magazine*, **12**(2), pp. 21—28, 2005.
- S. Nakrani and C. Tovey, on honey bees and dynamic allocation in an internet server ecology, in *Proc. 2nd Int'l Workshop on the Mathematics and Algorithms of Social Insects*, 2003.
- B. Niu, Y. Zhu, X. He, and X. Zeng, Optimum design of PID controllers using only a germ of intelligence, in *Proc. 6th World Congress on Intelligent Control and Automation*, pp. 3584—3588, 2006.
- Rafael S. Parpinelli, Heitor S. Lopes and Alex A. Freitas, Data Mining With an Ant Colony Optimization Algorithm, *IEEE Transactions on Evolutionary Computation*, **6**(4), pp. 321—332, 2002.
- Partridge, B.L., The Structure and Function of Fish Schools, *Scientific American*, June 1982, pp. 114—123.
- Kevin M. Passino, Biomimicry of Bacterial Foraging for distributed optimization and control, *IEEE Control Systems Magazine*, pp. 52—68, June, 2002.
- Pini G., Brutschy A., Birattari M., Dorigo M. Interference Reduction Through Task Partitioning in a Robotic Swarm, In *6th International Conference on Informatics in Control, Automation and Robotics (ICINCO 09)*, 2009
- Potts, W.K, The Chorus-Line Hypothesis of Manoeuvre Coordination in Avian Flocks, letter in Nature, Vol. **309**, May 24, 1984, pp. 344—345.
- Nicanor Quijano and Kevin M. Passino, Honey Bee Social Foraging Algorithms for Resource Allocation, Part I: Algorithm and Theory, in *Proc. 2007 American Control Conference*, pp. 3383—3388, 2007a.
- Nicanor Quijano and Kevin M. Passino, Honey Bee Social Foraging Algorithms for Resource Allocation, Part II: Application, in *Proc. 2007 American Control Conference*, pp. 3389—3394, 2007b.
- Margarita Reyes-Sierra and Carlos A. Coello Coello, Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art, *International Journal of Computational Intelligence Research*, **2**(3), pp. 287—308, 2006.
- Craig Reynolds, Flocks, Herds and Schools: A Distributed Behavioral Model, *Computer Graphics*, **21**(4), pp. 25—34, 1987.

- Roberts J., Zufferey J., Floreano D. Energy Management for Indoor Hovering Robots In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2008)*, IEEE (editors), September, 2008.
- Laura Rosati, Matteo Berioli and Gianluca Reali, On ant routing algorithms in ad hoc networks with critical connectivity, *Ad Hoc Networks*, **6**(6), pp. 827—859, 2008.
- Erol Sahin, Swarm Robotics: From Sources of Inspiration to Domains of Application, in *Swarm Robotics*, Springer LNCS 3342, pp. 10—20, 2005.
- R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, Ant-based load balancing in telecommunications networks. *Adaptive Behavior*, **5**(2):169—207, 1996.
- R. Schoonderwoerd, O. Holland, and J. Bruten. Ant-like agents for load balancing in telecommunications networks. In *Proceedings of the First International Conference on Autonomous Agents*, pages 209—216, ACM Press, 1997.
- J. Segall, S. Block and H. Berg, Temporal comparisons in bacterial chemotaxis, *PNAS*, **83**, pp. 8987—8991, 1986.
- X.H. Shi, Y.C. Liang, H.P. Lee, C.Lu, and L.M. Wang, An improved GA and a novel PSO-GA-based hybrid algorithm, *Information Processing Letters*, **93**:255—261, 2005.
- Krzysztof Socha and Marco Dorigo, Ant colony optimization for continuous domains, *European Journal of Operational Research*, **185**:1155—1173, 2008.
- W.J. Tang, Q.H. Wu, and J.R. Saunders, A Bacterial Swarming Algorithm For Global Optimization, in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 1207—1212, IEEE Service Center, Piscataway, NJ, 2007.
- Guy Theraulaz, Du super-organisme a l'intelligence en essaim: modeles et representations du fonctionnement des societes d'insectes, in *Intelligence Collective* (Eric Bonabeau and Guy Theraulaz, eds.), pp. 29—109, Paris: Hermes, 1994.
- Guy Theraulaz and Eric Bonabeau. Modelling the Collective Building of Complex Architectures in Social Insects with Lattice Swarms. *Journal of Theoretical Biology*, **177**(4), pp. 381—400, 1995.
- Guy Theraulaz, Eric Bonabeau, Stamatis C. Nicolis, Ricard V. Sole, Vincent Fourcassie, Stephane Blanco, Richard Fournier, Jean-Louis Joly, Pau Fernandez, Anne Grimal, Patrice Dalle and Jean-Louis Deneubourg.. Spatial patterns in ant colonies. *PNAS*, **99**(15), 2002.
- M. Tripathy and S. Mishra, Bacteria foraging-based solution to optimize both real power loss and voltage stability limit, *IEEE Transactions on Power Systems*, **22**(1), 2007.
- Frans van den Bergh and Andries-P. Engelbrecht, A Cooperative Approach to Particle Swarm Optimization, *IEEE Transactions on Evolutionary Computation*, **8**(3):225—239, 2004.
- Robert K. Vander Meer and Leanne E. Alonso, Pheromone Directed Behaviour in Ants, in *Pheromone Communication in Social Insects*, Vander Meer et al. (eds.), Westview Press, Boulder, CO, pp. 159--192, 1998a.
- Robert K. Vander Meer, M. Breed, M. Winston and K.E. Espelie (eds), *Pheromone Communication in Social Insects*, Westview Press, Boulder, CO, 368pp., 1998b.
- Waibel M., Keller L., Floreano D. (2009) Genetic Team Composition and Level of Selection in the Evolution of Multi- Agent Systems, *IEEE Transactions on Evolutionary Computation*, **13**(3): 648—660, 2009
- Reginald L. Walker, Dynamic Load Balancing Model: Preliminary Assessment of a Biological Model for a Pseudo-Search Engine, in *Parallel and Distributed Processing*, LNCS vol. 1800, Springer, pp. 620—627, 2000.
- Bo Yang, Yunping Chen and Zunlian Zhao, Survey on Applications of Particle Swarm Optimization in Electric Power Systems, in *IEEE Int'l Conf. on Control and Automation*, pp. 481—486, 2007.

- Peng-Yeng Yin, Fred Glover, Manuel Laguna, and Jia-Xian Zhu, Scatter PSO --- A More Effective Form of Particle Swarm Optimization, in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 2289—2296, 2007.
- Hua Yuan, Yi-Li, Wenhui Li, Kong Zhao, Duo Wang, and Rongqin Yi, *Combining Immune with Ant colony Algorithm for Geometric Constraint Solving*, in *Proceedings of the 2008 Workshop on Knowledge Discovery and Data Mining*, pages 524—527, IEEE Computer Society, 2008.
- Rui Zhang and Cheng Wu, An Effective Immune Particle Swarm Optimization Algorithm for Scheduling Job Shops, in *Proceedings of the 3rd IEEE Conference On Industrial Electronics and Applications*, pages 758—763, 2008.