Programming Nanotechnology: Learning from Nature

BOONSERM KAEWKAMNERDPONG

Department of Computer Science
University College London
UK

NAVNEET BHALLA AND PETER J. BENTLEY

Department of Computer Science
University of Calgary
Canada

Abstract

For many decades, nanotechnology has been developed with cooperation from researchers in several fields of studies including physics, chemistry, biology, material science, engineering, and computer science. In this chapter, we explore the nanotechnology development community and identify the needs and opportunities of computer science research in nanotechnology. In particular we look at methods for programming future nanotechnology, examining the capabilities offered by simulations and intelligent systems. This chapter is intended to benefit computer scientists who are keen to contribute their works to the field of nanotechnology and also nanotechnologists from other fields by making them aware of the opportunities from computer science. It is hoped that this may lead to the realisation of our visions.

ISSN: 0065-2458/DOI: 10.1016/S0065-2458(06)71001-4

1. Introduction

In 1959, Richard Feynman, a future Nobel Laureate, gave a visionary talk entitled "There's Plenty of Room at the Bottom"[1] on miniaturisation to nanometre-scales. Later, the work of Drexler [1,2] also gave futuristic visions of nanotechnology. Feynman and Drexler's visions inspired many researchers in physics, material science, chemistry, biology and engineering to become nanotechnologists. Their visions were fundamental: since our ancestors made flint axes, we have been improving our technology to bring convenience into our everyday life. Today a computer can be carried with one hand—40 years ago a computer (hundreds of times slower) was the size of a room. Miniaturisation of microprocessors is currently in process at nanometre-scales [3]. Yet, the style of our modern technology is still the same as ancient technology that constructed a refined product from bulk materials. This style is referred to as *bulk* or *top–down technology* [1]. As conventional methods to miniaturise the size of transistors in silicon microprocessor chips will soon reach its limit[2] and the modification of today's top–down technology to produce nanoscale structures is difficult and expensive [3], a new generation of computer components will be required. Feynman and Drexler proposed a new style of technology, which assembles individual atoms or molecules into a refined product [1]. This Drexler terms *molecular technology* or *bottom–up technology* [1]. This bottom–up technology could be the answer for the computer industry. Though top–down technology currently remains the choice for constructing mass-produced devices, nanotechnologists are having increasing success in developing bottom–up technology [3].

[1] For more information, see http://www.zyvex.com/nanotech/feynman.html.
[2] From http://science.howstuffworks.com/nanotechnology2.htm.

There are some concerns regarding emergent bottom–up technology. First, the laws of physics do not always apply at nanometre-scales [4]. The properties of matter at nanometre-scales are governed by a complex combination of classical physics and quantum mechanics [4]. Nevertheless, bottom–up fabrication methods have been successfully used to make nanotubes and quantum dots [3]. These methods are not yet suitable for building complex electronic devices such as computer processors, not to mention nanoassemblers that can make copies of themselves and work together at a task. Furthermore, and significantly, once knowledge of nanotechnology is advanced and real-world nanoassemblers are realised, they must be properly controllable to prevent any threats to our world.

More recently computer science has become involved in nanotechnology. Such research is wide ranging and includes: software engineering, networking, Internet security, image processing, virtual reality, human–machine interface, artificial intelligence, and intelligent systems. Most work focuses on the development of research tools. For example, computer graphics and image processing have been used in nanomanipulators that provide researchers an interactive system interface to scanning-probe microscopes, which allow us to investigate and manipulate the surface at atomic scales[3] [5,6]. In addition, genetic algorithms have been used as a method in automatic system design for molecular nanotechnology [7].

Computer science offers more opportunities for nanotechnology. Soft Computing techniques such as swarm intelligence, genetic algorithms and cellular automata can enable systems with desirable emergent properties, for example growth, self-repair, and complex networks.[4] Many researchers have successfully applied such techniques to real-world problems including complex control systems in manufacturing plants and air traffic control.[4] With some modifications towards nanotechnology characteristics, these techniques can be applied to control a swarm of a trillion nanoassemblers or nanorobots (once realised). It is anticipated that soft computing methods such as these will overcome concerns about implications of nanotechnology, and prevent the notorious scenario of self-replicating nanorobots multiplying uncontrollably.

This chapter reviews nanotechnology from different points of view in different research areas. We discuss the development of the field at the present time, and examine some concerns regarding the field. We then focus on the needs and benefits of computer science for nanotechnology, as well as existing and future computer science research for nanotechnology. The second half of this chapter introduces the area of swarm intelligence and then summarises investigations into how nanotechnology and self-assembling devices may be controlled by such techniques.

[3] For more information, see http://www.cs.unc.edu/Research/nano/cismm/nm/index.html.

[4] From http://www.nanotec.org.uk/evidence/92aUKCRC.htm.

2. Development in Nanotechnology

To describe Feynman's grand visions that have inspired many researchers in several fields of study, Drexler[5] introduced the term "Nanotechnology" and "Molecular Engineering" in his book, "Engines of Creation" [1]. He explored and characterised an extensive view of Feynman's visions in many aspects including potential benefits and possible dangers to humanity. According to the vision, building products with atomic precision by bottom–up technology could offer a dramatic widespread of potential and a decrease in environmental impact which would improve our way of life. A simple example of potential benefits from nanotechnology is that information stored on devices could be packed into much smaller spaces so that less pollution from discarding those devices would be produced. The aspect that would be directly beneficial to humankind is nanomedicine, which involves medical research at nanoscale [1,8]. For example, a group of programmable nanorobots that could flow along our bloodstreams without harm to our bodies could be injected to treat our bodies from within.

Nanotechnology has indeed promised a great future for humanity. However, the down side of the technology should not be neglected. Drexler suggested the potential threats to life on Earth of uncontrollably replicating assemblers [1]. In order to prevent any threat to society, it is crucial that nanotechnology is developed under acceptable standards with regard to ethical and social considerations. Recently, the Foresight Institute, which is a non-profit organisation for nanotechnology, gave version 4.0 of its guidelines as a self-assessment list for research and development in the field of nanotechnology.[6] The Science Media Centre has also produced a document describing nanotechnology for use by the media.[7] Today nanotechnology is gaining public attention. Many companies have been doing research and development in nanotechnology for commercial purposes. The governments of several countries have begun funding for research in this area. This recent development of nanotechnology is described further in the following sections.

2.1 Nanomanipulators

One important concept of nanotechnology is building products using bottom–up technology. Instead of sculpting bulk materials into desired products, bottom–up technology suggests a new method that assembles individual atoms into products. The first step to bottom–up technology is to acquire the ability to manipulate individual atoms at the scale of nanometres as desired. Therefore, the development of a

5 From http://www.foresight.org/FI/Drexler.html.

6 For more information, see http://www.foresight.org/guidelines/current.html#Principles.

7 For more information, see http://www.sciencemediacentre.org/nanotechnology.htm.

nanomanipulator, which is a tool for manipulating nanoscopic materials, is seen by some as being crucial to the progress of nanotechnology.

The first imaging in nanoscale was from the electron microscope developed by M. Knoll and E. Ruska in 1931 [9]. Later in 1981, G. Binnig and H. Rohrer invented the scanning tunnelling microscope (STM)[8] that can image individual atoms, and earned the Nobel Prize [9]. The success of the scanning tunnelling microscope leads to the development of other scanning probe microscopes (SPM) including the atomic force microscope (AFM). Instead of using lenses like traditional microscopes, all these scanning probe microscopes use a probe to scan atoms over the surface, measure a local property and result the image. Different types of scanning probe devices are designed for different tasks. For example, the STM is only appropriate when the material conducts electricity, while the AFM can work with non-conducting materials.

Apart from resembling a surface at atomic scale into a high-resolution image, scanning probe microscopes can be used to manipulate individual atoms. In 1990, D.M. Eigler of IBM used an STM to precisely place xenon atoms on a nickel plate into the name "IBM"[9] [10,2]. In 1993, W. Robinett and R.S. Williams developed a virtual reality system that allowed user to see and touch atoms via the scanning tunnelling microscope [9,11]. This was the beginning of a nanomanipulator.[10] At the University of North Carolina, another nanomanipulator has been developed in a multi-disciplinary project involving in the collaboration of several departments including computer science, physics and chemistry.[11] This nanomanipulator is a virtual-reality interface to scanning probe devices. Using technology in computer graphics, the features that are faint in the image can be enhanced. The system allows scientists to investigate and manipulate the surface of materials at the atomic scale. As a result, it has led to new discoveries in biology, material science and engineering.[11] For example, scientists have used the nanomanipulator system to examine the mechanical and electrical properties of carbon nanotubes [6]. Nanomanipulators are now commercially available. However, the ability to manipulate individual atoms alone could not yet enable us to build reliable nanomachines, unless the physical principles at nanoscales are comprehended.

[8] The STM can show surface topography imaging in high resolution by scanning its electrically conducting tip over a surface at the distance of a few atoms. The STM measures the electrical current called the *tunnelling current* between the tip and each point of the surface. As the distance between the tip and the surface changes, the tunnelling current is altered. The STM adjusts the vertical position of the tip to maintain a constant distance to the surface. These adjustments are recorded as a grid of values and finally transformed into image. For more information, see http://www.almaden.ibm.com/vis/stm/.

[9] See the picture at http://www.almaden.ibm.com/vis/stm/atomo.html#stm10.

[10] From http://www.cs.unc.edu/Research/nano/cismm/nm/index.html.

[11] For more information, see http://www.cs.unc.edu/Research/nano/cismm/nm/.

2.2 Nanofabrication

After scientists have gained the ability to manipulate individual atoms directly, the next step is to manufacture structures at nanometre scale, i.e., structures smaller than 100 nanometres across. In this section, we discuss nanofabrication methods, which can be divided into two categories: top–down methods and bottom–up methods [3]. Akin to the concept of technology styles discussed previously, the top–down methods involve carving out or adding a small number of molecules to a surface, while the bottom–up methods assemble atoms or molecules into nanostructures.

A top–down method that has been used in the electronics industry is photolithography. Photolithography is the process that transfers the geometric shape on a mask to the surface of a silicon wafer by exposure to UV light through lenses. The computer industry uses this technology to fabricate microprocessor chips [3]. However, the use of photolithography to fabricate nanostructures is limited by the wavelength of the UV light. One modification can be made by using electron-beam lithography, which is a technique for creating fine patterns on a thin polymer film with a beam of electrons [3,12]. Because electron-beam lithography is very expensive and slow, the development of soft lithography, which is a process that creates elastic stamp in order to transfer structures to a surface, allows researchers to reproduce patterns inexpensively in a wide range of materials. Nevertheless, this technique is not yet ideal for manufacturing complex multi-layered structure electronic devices. The need for methods to fabricate complex nanostructures that are simpler and less expensive has stimulated researchers to explore unconventional approaches.

Another top–down method involves using the scanning probe microscopes that were used to manipulate individual atoms to spell IBM. Researchers can manipulate atoms with an STM in three modes: pushing, pulling and sliding. Apart from mechanical manipulation, the STM can be used to assist in fabrication by chemistry catalysing. In 1995, W.T. Muller et al. proposed a method to use scanning probe microscopes in nanofabrication [13]. They used a platinum-coated AFM tip to scan over the surface coated with a monolayer of azide ($–N_3$) compounds. As a result, amino groups are formed by catalytic conversions of azide and can be used to generate more complex structures. Another nanofabrication method using scanning probe devices was introduced by E.S. Snow and P.M. Campbell [14]. Their technique was to add a bias current to the AFM tip and monitor the electrical resistance of the structure during the fabrication process. When the target resistance is reached, the bias is switched off. This innovative feedback mechanism has been modified and used in later research. Recently, F. Rosei et al. proposed a novel nanofabrication method for metal structures [15]. This method uses organic molecules as templates for the rearrangement of copper atoms on a surface. At low temperature where the copper atoms are static, the template molecules can be moved away without damaging the copper surface by precisely controlling the STM tip. For more information on using the scanning probe devices in fabrication, a review by S.W. Hla and K.H. Rieder is recommended[12] [16].

In contrast, bottom–up methods are truly representing a new style of technology. Although the advancement of the bottom–up methods may not yet be suitable for the production of electronic devices or allow us to replace conventional top-up methods in fabrication, researchers can inexpensively assemble atoms and molecules into nanostructures with dimensions between 2 and 10 nanometres by self-assembly chemical reactions. One innovation created with a bottom–up method is a carbon nanotube discovered by S. Iijima of NEC in 1991 [17,9]. A carbon nanotube is a tube-shaped carbon material that is measured in nanometre scales. It became the fifth type of solid-state carbon[13] after diamond structures, graphite structures, non-crystalline structures and fullerene molecules or buckyballs, which were discovered by R.F. Curl et al. in 1985 [9]. Since then, researchers have been studying the properties and characteristics of carbon nanotubes. Different structures of carbon nanotubes varying in length, thickness, type of spiral and number of layers have been developed for various purposes. Recently, NEC announced the world's first compact fuel cell for mobile devices that uses spiral-shaped carbon nanotubes or nanohorns for the electrodes.[14] Carbon nanotubes are expected to be a key material in the future.

Another new material, quantum dots, is made by bottom–up methods. Quantum dots are crystals that emit only one wavelength of light when their electrons are excited. Because the electrical, magnetic and optical properties of the dot are regulated by the size of the dot, the production of quantum dots must maintain their size and composition [3]. The size of the dots can be selected by varying the amount of time for the chemical reaction. The emission of light by quantum dots could be used in medicine as a biological marker [3]. Alternatively, quantum dots could be used as quantum bits and to form the basis of computers.[15]

2.3 Nanocomputers

In 1965, G. Moore, the co-founder of Intel, predicted a trend that the number of transistors contained in a microprocessor chip would double approximately every 18 months. This became known as Moore's law. As exemplified in Intel's chips,[16] the prediction appears surprisingly correct. However, without the development of nanotechnology researchers will struggle to meet the prediction of Moore's law.

[12] From http://www.imm.org/Reports/Rep040.html.

[13] For more information, see http://www.labs.nec.co.jp/Eng/innovative/E1/02.html.

[14] For more information, see http://www.smalltimes.com/document_display.cfm?document_id=7563.

[15] For more information, see http://news.uns.purdue.edu/html4ever/010917.Chang.quantum.html.

[16] For more information, see http://www.intel.com/research/silicon/mooreslaw.htm.

One of the first achievements in nanocomputer research was perhaps the development of single-electron tunnelling (SET) transistors by D. Averin and K. Likharev in 1985.[17] Later in 1987, T.A. Fulton and G.J. Dolan at Bell Laboratories fabricated single-electron transistors and made an observation on the quantum properties and effects of electrons when transistors are in operation [18]. As techniques in nanofabrication advances, researchers have successfully created electronic components including transistors, diodes, relays and logic gates from carbon nanotubes [19,20]. The next step is providing the interconnection between components. Researchers have been working on a different type of nanoscale wire called a semiconductor nanowire and studied how to interconnect and integrate the components [19]. The final step to build a computer processor is to fabricate the designed circuit. Recently, the semiconductor industry has successfully built 70-megabit memory chips containing over half billion transistors.[18] As the advancement in nanofabrication progresses, the silicon-based nanocomputer becomes closer to reality.

Another approach to nanocomputers is DNA computing. Deoxyribonucleic acid (DNA) is a nucleic acid that carries genetic information for the biological development of life. In 1994, L. Adleman introduced the idea of solving a well-known complex mathematical problem, called the travelling salesman problem, by using DNA [21]. His DNA computer showed that DNA could indeed be used to calculate complex mathematics; however, it is not yet comparable to conventional computer in terms of speed and ease of use. Nevertheless, his work has encouraged the development in DNA computing. In 1997, researchers at the University of Rochester built DNA logic gates, another step towards a DNA computer. The fact that a DNA molecule can store more information than any conventional memory chip and that DNA can be used to perform parallel computations make the area very appealing.[19] Regardless of the success of DNA computers, the development of silicon-based nanocomputers could use the advantages of DNA computing.

Apart from silicon-based nanocomputers and DNA computers, researchers believe that quantum computers may be another promising approach that overcomes the limits of conventional computers [22]. Feynman began one of the first research groups to explore computational devices based on quantum mechanics. In 1982, he demonstrated how computations could be done by quantum systems according to the principles of quantum physics [23]. In quantum computers, the binary data in conventional computers are represented by quantum bits, or qubits, which can be in a state of 0, 1 and superposition (simultaneously both 0 and 1). As a quantum computer can hold multiple states simultaneously, it is argued that it has the potential to

[17] For more information, see http://physicsweb.org/articles/world/11/9/7/1.
[18] From http://www.smalltimes.com/document_display.cfm?section_id=39&document_id=8257.
[19] From http://www.news.wisc.edu/view.html?id=3542.

perform a million computations at the same time.[20] However, quantum computers are based on quantum-mechanical phenomena, which are vulnerable to the effects of noise. A scheme for quantum error correction is required.[21] Researchers have been working to overcome this obstacle. To date, quantum computing is still in the very early stages.

2.4 Nanorobots

One vision of a nanoassembler or nanorobot is a device with robotic arms, motors, sensors and computer to control the behaviour, all at the scale of nanometres. In 1992, the book called "Nanosystem" by Drexler gives an analysis of the feasibility of machine components for such nanorobots [24]. However, even to build a molecular motor, researchers have to consider laws of thermodynamics when motors are actually in operation [25]. Just building a miniature version of an ordinary motor is not adequate. Recently, a controversy arose surrounding Feynman's vision of nanorobots. In 2003, an open debate through letters between K.E. Drexler and R.E. Smalley (who was awarded a Nobel Prize for the discovery of fullerenes) was presented to public.[22] Smalley was not convinced that such molecular assemblers envisioned by Drexler are physically possible, while Drexler insists on his previous findings. Certainly, the study of similarly-sized biological machines—organic cells—suggests there may be more effective alternatives to Drexler's nanorobots. Even if nanorobots can be realised, they will not be available in the near future [26].

2.5 Nanomedicine

Nanotechnology promises a great future for medical research including improved medical sensors for diagnostics, augmentation of the immune system with medical nanomachines, rebuilding tissues, and tackling aging. Proponents claim that the application of nanotechonology to medicine, so-called nanomedicine, offers ultimate benefits for human life and society by eliminating all common diseases and all medical suffering.[23] Eventually, it is argued that nanomedicine would allow the extension of human capabilities.[23] In 2003, R.A. Freitas Jr. commented that nanometre-scale structures and devices held great promises for the advancement of medicine includ-

[20] For more information, see http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume4/hogg96a-html/node6.html.

[21] For more information, see http://www.theory.caltech.edu/~quic/errors.html.

[22] The details of those letters can be found at http://pubs.acs.org/cen/coverstory/8148/8148counterpoint.html.

[23] From http://www.foresight.org/Nanomedicine/NanoMedFAQ.html#FAQ19.

ing advanced biosensors, smart drugs and immunoisolation therapies.[24] In this initial stage of nanomedicine, nanostructured materials are being tested in various potential areas; for example, tagging nanoparticles using quantum dot nanocrystals as biological markers and smart drugs that become active only in specific circumstances.[25] In addition, researchers have found a method to control the size of densely packed DNA structures, one of nature's efficient ways for transporting gene information.[26] This could improve the efficiency of gene therapy for medical treatment and disease prevention.[26] It is hoped by many that the next stage of nanomedicine, where nanorobots or nanocomputers are fully available, would expand enormously the effectiveness, comfort and speed of future medicine treatments with fewer risks and costs.

3. Benefits of Computer Science for Nanotechnology

Recently, M.C. Roco of the National Nanotechnology Initiative (NNI), an organisation officially founded in 2001 to initiate the coordination among agencies of nanometre-scale science and technology in the USA, gave a timeline for nanotechnology to reach commercialisation.[27] For the next twenty years, the NNI has divided the development of nanotechnology into four generations. The first generation, which just ended in 2004, involved the development of *passive nanostructures* such as coatings, nanoparticles, nanostructured metals, polymers and ceramics. At the time of writing, we begin the second generation, during which we should manufacture *active nanostructures* including transistors, amplifiers, targeted drugs, actuators and adaptive structures. Later, from the year 2010, nanotechnology should enter the third generation. It is estimated that *systems of nanosystems*, for example: guided molecular assembling systems, 3D networking and new system architectures for nanosystems, robotics and supramolecular devices, would be developed. Finally, from the year 2020, the fourth generation of nanotechnology should be the generation of *molecular nanosystems*, which would integrate evolutionary systems to design molecules as devices or components at atomic levels.

To date, nanotechnology has been developed mostly from the basis in physics, chemistry, material science and biology. As nanotechnology is a truly multidisciplinary field, the cooperation between researchers in all related areas is crucial to

[24] For more information, see http://www.nanotech-now.com/products/nanonewsnow/issues/003/003.htm.

[25] For more information, see http://www.sciencenews.org/articles/20040501/fob1.asp.

[26] From http://www.azonano.com/details.asp?ArticleID=104.

[27] The presentation material in his talk at the workshop Nanotechnology Research Direction II and other NNI presentation materials can be found at http://www.nsf.gov/crssprgm/nano/reports/nnipres.jsp.

the success of nanotechnology. Until now, computer science has taken a role mostly in research tools, for example: a virtual-reality system coupled to scanning probe devices in nanomanipulator project. However, according to M.C. Roco, the third and fourth generation of nanotechnology would rely heavily on research in computer science.

Perhaps reflecting the extensive use of computers in the modern world, computer science is today a broad field, with many aspects that may affect nanotechnology. Earlier sections have outlined the use of graphics and imaging with nanomanipulators. Other current uses of computer science for nanotechnology include developing software systems for design and simulation. A research group at NASA has been developing a software system, called *NanoDesign*, for investigating fullerene nanotechnology and designing molecular machines.[28] The software architecture of NanoDesign is designed to support and enable their group to develop complex simulated molecular machines.

However, here we focus on intelligent systems. Research in intelligent systems involves the understanding and development of intelligent computing techniques as well as the application of these techniques for real-world tasks, often including problems in other research areas. The techniques in intelligent systems comprise methods or algorithms in artificial intelligence (AI) including knowledge representation/reasoning, machine learning and natural computing or soft computing.

An exciting new development at the time of writing is a project called PACE (programmable artificial cell evolution). This large interdisciplinary project aims to create a "nanoscale artificial protocell able to self-replicate and evolve under controlled conditions."[29] The protocells in this work are intended to be the "simplest technically feasible elementary living units (artificial cells much simpler than current cells),"[30] These are intended to act as nanorobots, comprising an outer membrane, a metabolism, and peptide-DNA to encode information. Evolutionary modelling is being used extensively in PACE, to analyse real and simulated protocell dynamics, their possible evolution, and the evolution of (potentially noisy) protocellular networks. Evolution is also being used within microfluidic FPGA chips to produce stable self-replicating cell-membranes, with a genetic algorithm using physical populations on the chip and evaluated by a computer vision system. In addition to this work, computer modelling of embryogenesis and developmental systems is becoming increasingly popular in computer science [31]. Should artificial cells become a

[28] From http://www.nas.nasa.gov/Groups/Nanotechnology/publications/MGMS_EC1/NanoDesign/article.html.

[29] From http://complex.upf.es/~ricard/PACEsite.html.

[30] From http://134.147.93.66/bmcmyp/Data/BIOMIP/Public/bmcmyp/Data/PACE/Public/paceprosheet.html.

reality, such models will provide a method for their genes to be programmed in order to enable the growth of larger, multicellular forms.

Apart from genetic algorithms and other evolutionary algorithms that have promising potential for a variety of problems (including automatic system design for molecular nanotechnology [7]), another emerging technique is swarm intelligence, which is inspired by the collective intelligence in social animals such as birds, ants, fish and termites. These social animals require no leader. Their collective behaviours emerge from interactions among individuals, in a process known as self-organisation. This collective intelligence in social animals often cannot emerge from direct interaction among individuals. Instead, indirect social interaction (*stigmergy*) must be employed. Each individual may not be intelligent, but together they perform complex collaborative behaviours. Typical uses of swarm intelligence are to assist the study of human social behaviour by observing other social animals and to solve various optimisation problems [27,28]. There are three main types of swarm intelligence techniques: models of bird flocking, the ant colony optimisation (ACO) algorithm, and the particle swarm optimisation (PSO) algorithm. Different techniques are suitable for different problems.

Although still a young field of computer science, swarm intelligence is becoming established as a significant method for parallel processing and simultaneous control of many simple agents or particles in order to produce a desired emergent outcome. For example, researchers at the Santa Fe Institute developed a multi-agent software platform, called *Swarm*[31] inspired by collaborative intelligence in social insects, for simulating complex adaptive systems. Likewise, BT's Future Technologies Group developed a software platform known as EOS, for Evolutionary Algorithms (EAs) and ecosystem simulations. The group uses EOS for research into novel EAs and ecosystem models and for rapid development of telecommunication-related applications [32]. Systems such as these will become increasingly important for modelling molecular machine systems.[32] They are also being investigated as a solution to provide self-healing, adaptive and autonomous telecommunications networks. Another potential benefit of such techniques for complex adaptive systems in this area would be to control intelligently the manufacture of nanometre-scale devices, where no exact mathematical model of the system exists. Many intelligent systems' techniques have been successfully applied in control system of various complex applications. Although at nanometre-scale the principles and properties of materials are altered, researchers have attempted to solve other dynamic problems using soft computing techniques and have been developing new techniques to cope with such problems.

[31] For more information, see http://www.swarm.org/.

[32] From http://www.foresight.org/Updates/Update39/Update39.5.html.

Also inspired by emergent collaborating behaviours of social insects, the *Autonomous Nanotechnology Swarm* (ANTS)[33] architecture for space exploration by NASA Goddard Space Flight Center is claimed to be a revolutionary mission architecture. The ANTS architecture distributes autonomous units into swarms and organises them in hierarchy by using the concept of artificial intelligence. Researchers at the center have been developing a framework to realise the autonomous intelligent system by using an Evolvable Neural Interface (ENI). As a result, the interface allows cooperation between higher-level neural system (HLNS) for elementary purpose actions and lower-level neural system (LLNS) for problem solving as required in real-world situations. In the plan, each autonomous unit will be capable of adapting itself for its mission, and the ANTS structures will be based on carbon nanotube components.

In 1996, O. Holland and C. Melhuish investigated the abilities of single and multiple agents on a task with agents under similar circumstances as future nanorobots (minimal sensing, mobility, computation and environment) [29]. The task to be solved by the agents in their studies was to learn to move towards a light source by using simple rule-based algorithms. In the case of single agents, the result was efficient, but performance degraded as the amount of noise increased. In the case of multiple agents, the best result was from the algorithm that formed collective behaviours akin to genuine social insects. This investigation showed that emergent collective intelligence from social interactions among agents modelled on social insects could cope with the limited capabilities that would be inevitable in future nanoscale robots.

Recently, B. Kaewkamnerdpong and P.J. Bentley proposed a new swarm algorithm, called the *Perceptive Particle Swarm Optimisation* (PPSO) algorithm [30]. The PPSO algorithm is an extension of the conventional PSO algorithm for applications in the physical world. By taking into account both the social interaction among particles, and environmental interaction, the PPSO algorithm simulates the emerging collective intelligence of social insects more closely than the conventional PSO algorithm; hence, the PPSO algorithm would be more appropriate for real-world physical control problems. This is the first particle swarm algorithm to be explicitly designed with nanotechnology in mind. Because each particle in the PPSO algorithm is highly simplified (each able to detect, influence or impact local neighbours in limited ways) and the algorithm is designed for working with a large number of particles, this algorithm would be truly suitable for programming or controlling the agents of nanotechnology (whether nanorobots, nanocomputers or DNA computers), whose abilities are limited, to perform effectively their tasks as envisioned. Further details of this method are provided in the second part of this article.

This is seen as a crucial “missing link” in bottom–up nanotechnology: the control of the nanosized agents. A billion (or trillion) tiny particles, whether complex

[33] For more information, see http://ants.gsfc.nasa.gov/.

molecules or miniature machines, must all cooperate and collaborate in order to produce the desired end result. None will have, individually, sufficient computing power to enable complex programming. Like the growth of crystals, the development of embryos, or the intelligent behaviour of ants, bottom–up nanotechnology must be achieved through collective, emergent behaviours, arising through simple interactions amongst itself and its environment. Computer science, and especially fields of research such as swarm intelligence, will be critical for the future of bottom–up nanotech.

We now examine swarm intelligence more closely and provide some examples of how it may be used for nanotechnology.

4. Swarm Intelligence

Swarm intelligence is inspired by collaborative behaviours in social animals such as birds, ants, fish and termites. Collaborative behaviour among social animals exhibits a remarkable degree of intelligence. These social animals require no leader. Their collaborative behaviours emerge from interactions among individuals. Often the behaviour of flocks, swarms and insect colonies, arises through interactions among the individuals in the group and through interactions with their environment. For example, ant foraging behaviour in many ant species arises by means of attractive pheromone trail [28].

Forager ants lay pheromone trail as they move from a food source to their nest. The other foragers sense and, then, follow the trail to the food source. Pheromone—which is a chemical substance—deposited in the environment serves as an intermediate agent in indirect interactions among individuals. Similarly, termites construct their mound by depositing the pheromone and following the smell of pheromone [27]. Individual termites move towards the direction with strongest pheromone concentration and deposit a mixture of local soil and their saliva. With such simple activities, termites construct their mound to fill with chambers, passages, and ventilation system even though they have no construction plan beforehand. Termites are considered to be some of the greatest architects in the insect (and animal) world [44]. Although each individual in insect colonies may not be intelligent, together they perform complex collaborative behaviours. Swarm intelligence techniques model collective behaviours in social insects.

4.1 Stigmergy and Self-Organisation

Unlike the hierarchical organisation with centralised control in humans (i.e., our brain is in one place and controls everything), social insects have no leader to coordinate other individuals to achieve their tasks; the role of so-called *queen* in insect colonies is merely a reproducer. With simple behaviours like pheromone laying and following, worker termites—which are blind—can build their sophisticated mound. Such indirect form of communication found in social insects is known as *stigmergy*. In 1959, Pierre-Paul Grasse made observations on termite building behaviour and used this term to describe task coordination and construction regulation in termites [28]. Grasse explained that the workers were guided by the construction [45]; individual worker deposits a chunk of material that stimulates the same worker or any other workers nearby to respond and deposit more material [28]. In general, stigmergy describes the indirect communication among individuals through the environment [27]; one individual modifies the environment, and the other individuals respond to the changed state of the environment leading to collaborative behaviours as seen in ants, termites, and other social insects.

In 1990, Deneubourg and his colleagues [46] showed that simulated robots with stigmergy and simple rules could achieve clustering and sorting tasks that are common activities in ants; some species of ants cluster corpses of their nestmates into cemetery and sort their larvae into piles according to size [28]. Without direct communication among robots, ant-like robots comprising with a short-term memory unit perform both tasks comparable to ants [46]. Deneubourg et al., however, note that long memory length prevents effective performance in clustering and sorting. In 1994, Beckers, Holland and Deneubourg [47] conducted similar clustering experiments on physical mobile robots with no memory [45]. The study shows that stigmergy can control and coordinate a number of robots to achieve their tasks and the number of robots is a critical factor to the performance of the system [47]. Whereas a greater number of robots reduces the time to achieve the task, increasing number of robots results in the exponentially increased number of interactions and may lead to the destruction of existing clusters [47].

Experiments on physical robots for sorting tasks were extended in [45] in 1999. Holland and Melhuish explored the effect of increasing number of agents in more details. Both direct and indirect interactions among social insects are required in the underlying mechanism to their collaborative behaviours, which is known as *self-organisation* [28]. Self-organisation is initially used to describe the mechanism of macroscopic patterns emerging from processes and interactions at microscopic level [48]. Likewise, the emergence of collective intelligence at colony level in social insects arising from the interactions among individuals with simple behaviours is from self-organisation as well. Bonabeau et al. [28] describes that self-organisation relies on four fundamental components:

1. *Positive feedback*: In foraging, when the individual comes back from the food source, it can recruit the others to this food source (either by dancing in bees

or pheromone trail in some ant species). The recruitment of the individuals and reinforcement are forms of positive feedback.

2. *Negative feedback*: To stabilise the collective pattern (i.e., suppress the positive feedback), negative feedback may be in the form of saturation, exhaustion, or competition.
3. *Amplification of fluctuations*: These fluctuations are, for instance, errors, random task-switching, and so forth. Randomness can promote the exploration and discovery of new solutions.
4. *Multiple interactions*: Self-organisation relies on multiple interactions including both direct and indirect interactions among individuals. For example, the action of pheromone-following can interact with pheromone-laying action if the density of the pheromone is sufficient. The pheromone substance can, however, evaporate over time; multiple interactions are required to maintain the pheromone density level and, hence, self-organisation.

The collective patterns and behaviours arising from self-organisation may not be completely orchestrated; termites do not know the order of activities they should do or specific location they should deposit soil to construct their mound. Nevertheless, stigmergy provides flexibility and robustness. Social insects can often collectively cope with external perturbation to their systems and exhibit the same collaborative behaviours [28]. Therefore, artificial agents adopting stigmergy and self-organisation can respond to perturbation without reprogramming [28]. Such intelligence can be transformed into a powerful tool in computer science.

4.2 Swarm Intelligence Techniques

The term "swarm intelligence" was first used to describe self-organised cellular robotic systems using nearest-neighbour interactions in [49,50]. Bonabeau et al. [28] later extended the definition to include: "any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies." As the intelligence in social animals emphasises decentralisation, self-organisation, direct/indirect interactions among simple agents, flexibility, and robustness, swarm intelligence techniques are simulated model of such intelligence and typically used to solve optimisation problems [27]. In swarm intelligence, there are two main types of techniques: *ant colony optimisation* (ACO) algorithm and the *particle swarm optimisation* (PSO) algorithm. These techniques are described as follows.

4.2.1 Ant Colony Optimisation

Inspired by foraging behaviour of ants, ant colony optimisation (ACO) algorithms are probabilistic-based computational methods modelling such collective behaviour in ants [51,52]. When foraging, individual ants randomly travel to find food source and lay a certain amount of pheromone on the way back from food source. Ants that sense attractive pheromone follow the trail left by other ants to the food source. When more than one trail is found, the one with stronger pheromone is more preferable and foragers are recruited to that more attractive trail; entomologists have shown that ants probabilistically prefer the path holding high pheromone concentration [53]. With this simple behaviour, ants can find the shortest path from their nest to a food source. Apart from foraging, some ant species cluster their dead to clean the nest in similar way.

The pheromone substance laid by individual ants plays an important role in locally indirect communication among individual ants in the neighbourhood—within the proximity of pheromone trail to detect pheromone concentration. After some time, the most promising path to food source has the greater pheromone concentration. ACO uses this positive feedback mechanism to reinforce the system to good solutions as the increasing amount of pheromone reflects recruitment and reinforcement on the solution.

The persistent pheromone allows ACO to keep good solutions in memory and to find better solutions [28]. As pheromones evaporate over time, the pheromone evaporation serves as negative feedback to avoid premature convergence. The pheromone evaporation and probabilistic randomness in artificial ants allow the ants to explore new paths. To balance between exploitation of the current food source—solution—and exploration of new solution, the rate of pheromone evaporation must be appropriately set. If the virtual pheromone evaporates too quickly, no collective behaviour can emerge; on the other hand, if it evaporates too slowly, the system can yield premature convergence [28].

The ACO framework, or called *ACO Meta-Heuristic*, apply this collective behaviour to solve combinatorial optimisation problems [28]. The examples of combinatorial optimisation problems—optimising with qualitative variables—include travelling salesman problem (TSP), quadratic assignment problem, and telecommunication network routing. In 1997, Dorigo and Gambardella demonstrated the use of ACO in travelling salesman problem where the shortest path length to visit all cities is required [54]. For TSP problem of 50 cities, ACO yields comparable results to other methods in the literature including simulated annealing, neural network, genetic algorithm, and farthest insertion heuristic; ACO often produces better results than the others [54].

Even though ACO algorithms are powerful optimisation methods and can be applied to both discrete and continuous optimisation problems, they have a limitation to be employed in physical applications. As pheromone is the essence of ACO metaheuristic, there must be real pheromone (or other substances that would serve a similar purpose) and an appropriate environment in which pheromone is deposited. For the task of three-dimensional nanorobot coordination control, ACO is only likely to be suitable for very specialised types of application (e.g., laying down of conductive paths between electronic components).

4.2.2 Particle Swarm Optimisation

Self-organisation in bird flocking is one of intriguing phenomena in nature. A large number of birds can flock synchronously, often change direction spontaneously, sometimes scatter and, then, regroup. Bird flocking behaviour has been studied for the underlying mechanisms in their social behaviours [55]. Scientists have developed computer simulations of the movement of social animals like bird flocks and fish schools [56]. One motive in developing such simulations was to model human social behaviour [57]. In 1987, Craig Reynolds developed a model of motion in social animals such as birds and fish [58]. His flocking model applies three simple behaviours to control the movement of simulated creature, called a *boid*. Each boid observes its neighbours, which are boids locating within the defined distance from itself, and acts according to three behaviours: avoiding collisions with its neighbours, matching velocity with its neighbours, and staying close to its neighbours. Using the combination of simple behaviours, the model shows that group behaviours can arise from interactions among boids within the neighbourhood. This model has been used in a simulation a swarm of bats in the film ”Batman Returns” in 1992[34] and other applications in computer animation and behavioural simulation [59].

Rather than relying on manipulations for optimum distances amongst individuals as in [56], Kennedy and Eberhart simulate human social behaviours according to sociobiologist E.O. Wilson [60] that social sharing of information among individuals offers an evolutionary advantage [57]. Through simulation, they discovered and developed an optimisation method for continuous non-linear functions called *particle swarm optimisation* (PSO) [57]. The original PSO algorithm resembles swarm intelligence through a very simple concept whose implementation requires inexpensive computation speed and memory requirement as the algorithm uses only primitive mathematic operators [57]. PSO algorithm is a population-based technique exhibiting self-organisation through social interactions and the exchange of information which each individual experiences; a swarm of particles fly around the problem space to find a good solution (position) with the influences from their own experience and their neighbours' experiences. Referring back to the self-organisation

[34] From http://www.red3d.com/cwr/boids/.

principles listed earlier, these experiences (or knowledge) from multiple social interactions according to neighbourhood topology can serve as positive feedback to influence particles to move towards a good position which may be, perhaps, an optimum. Meanwhile, these influences can serve as negative feedback as well; when a better position or optimum is found, social knowledge drives particles to leave the current optimum to pursue the better one. In any case, particles do not move directly towards a good position but rather explore around the good position as the PSO algorithm adds randomness in particle movement. Social interaction among individuals is crucial to the success of the PSO algorithm. In the conventional PSO algorithm, the exchange of information seems to arise from direct communication among particles and no stigmergy is regarded.

Since it was first introduced, the PSO algorithm has been continually modified to improve its performance to solve numerical optimisation problems. The particle swarm optimisation algorithm has been successfully employed to solve a range of optimisation problems including electric power systems [61], music [62], image classification [63], logic circuit design [64], recommender systems [65] and enhancement of other learning algorithms [66].

Using a similar representations to physical agents, the PSO algorithm seems a plausible method for application in physical applications including nanorobot coordination control. The inexpensive requirement in memory and computation suits well with nanosized autonomous agents whose capabilities may be limited by their size. Nevertheless, the conventional PSO algorithm requires complex, direct communication among particles in the neighbourhood which might not be possible in nanorobot. To apply in nanorobot control, a modification of PSO algorithm is required.

5. Perceptive Particle Swarm Optimisation

In particle swarm optimisation, all individuals in the swarm have the same behaviours and characteristics. It is assumed that the information on the position and the performance of particles can be exchanged during social interaction among particles in the neighbourhood. Importantly, conventional particle swarm optimisation relies on social interaction among particles through exchanging detailed information on position and performance. However, in the physical world, this type of complex communication is not always possible. Global communication may be impossible amongst swarm of nanorobots. Indeed, it is common for macro-sized robots to have no idea of their own performance in a given location and thus there may be little direct information that one individual can pass on to its companions.

Insects must cope with similar problems. Termites do not build their mounds by talking to each other and telling each other where to deposit material. Instead, they perceive each other, and they perceive their environment, and their complex behaviour emerges as a result of those perceptions. There is no concept of communication, only interaction. Social interaction and environmental interaction (stigmergy) enables termites to build highly complex structures without any direct communication.

Recent work by the authors has focused on the use of swarm intelligence for physical nanotechnology applications, where these kinds of severe communication restrictions are common. In order to imitate the physical collective intelligence in social insects, we have proposed the Perceptive Particle Swarm Optimisation (PPSO) algorithm, which adds an extra dimension to the search space and enables both social interaction and environmental interaction by allowing a finite perception range for each individual [27].

The PPSO algorithm is relatively similar to the conventional particle swarm optimisation algorithm. However, instead of operating in n-dimensional search spaces for n-dimensional optimisation problems, the PPSO algorithm operates in $(n + 1)$-dimensional search space. In effect, the particles fly over a physical fitness landscape, observing its peaks and troughs from afar. Instead of directly exchanging information among particles in their neighbourhoods, each individual has a finite range of perception so that it can observe the search space, which is the environment of the swarm, and perceive the approximate positions of other individuals within its perception range as social insects observe the world and other individuals through senses. Thus, particles in the PPSO algorithm are attracted to the better positions in the search space they perceive and to the neighbours they perceive.

The added dimension represents the underlying performance of particles at their positions in n-dimensional space. The exact performance at a specific position in the space is unknown to the particles in the PPSO algorithm. Adding the additional dimension and the ability to observe the search space allows particles to perceive their approximate performance. Consider an n-dimensional function optimisation problem. In the conventional particle swarm optimisation, particles fly around the n-dimensional search space and search for position giving the greatest performance measured by using the function to optimise. On the other hand, in the PPSO algorithm the particles fly around $(n + 1)$-dimensional space to observe the space and find the optima of the landscape. Because particles can fly "over" discontinuities and noise, the PPSO algorithm finds a good solution to the problem regardless of non-deterministic functions or stochastic conditions. Figure 1 shows particles (red dots) in the conventional PSO algorithm and the PPSO algorithm. In Fig. 1(a) and (b), particles operate in a one-dimensional problem, while Fig. 1(c) and (d) demonstrate particles operating in a two-dimensional problem.

In more detail: particles in the PPSO algorithm observe the search space within their perception ranges by sampling a fixed number of directions to observe and sampling a finite number of points along those directions. Figure 2 shows an example of

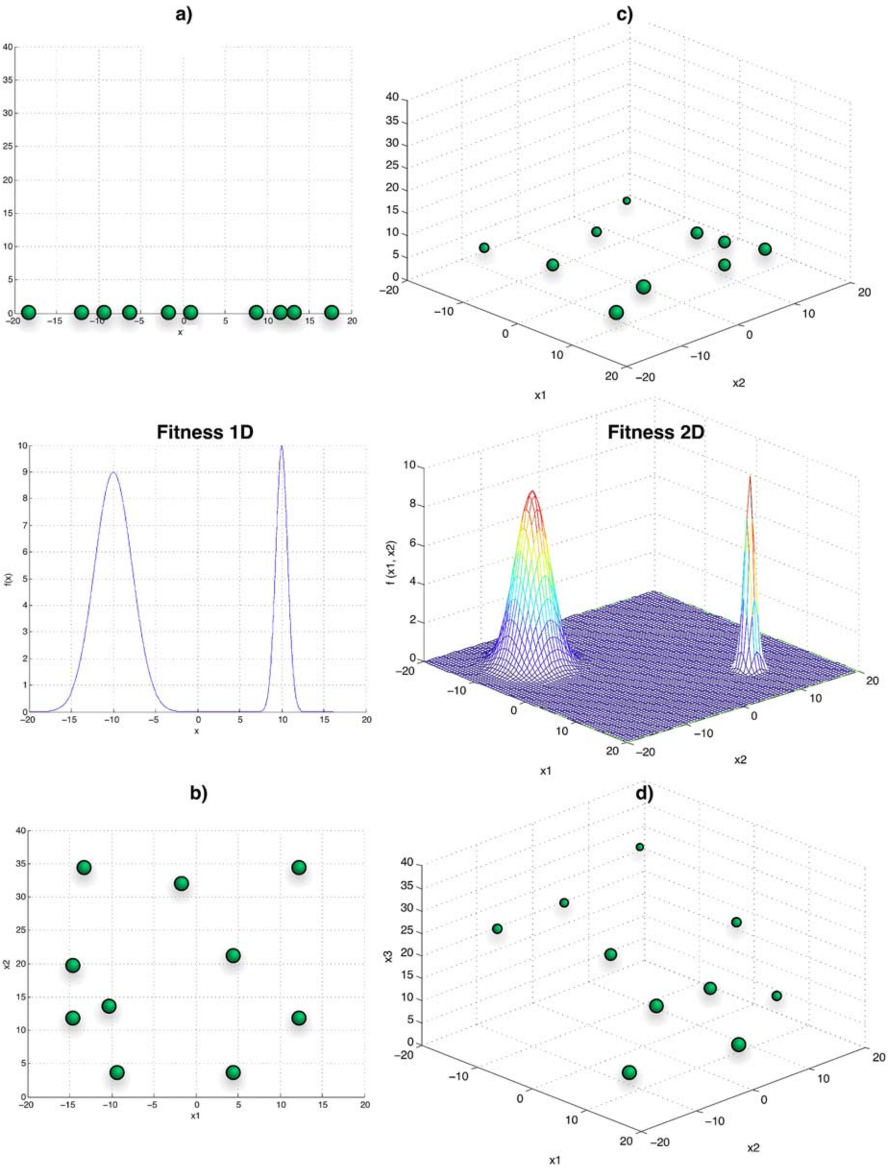


FIG. 1. The comparison between the conventional PSO algorithm (a), (c) and the PPSO algorithm (b), (d) in one-dimensional and two-dimensional optimisation problems.

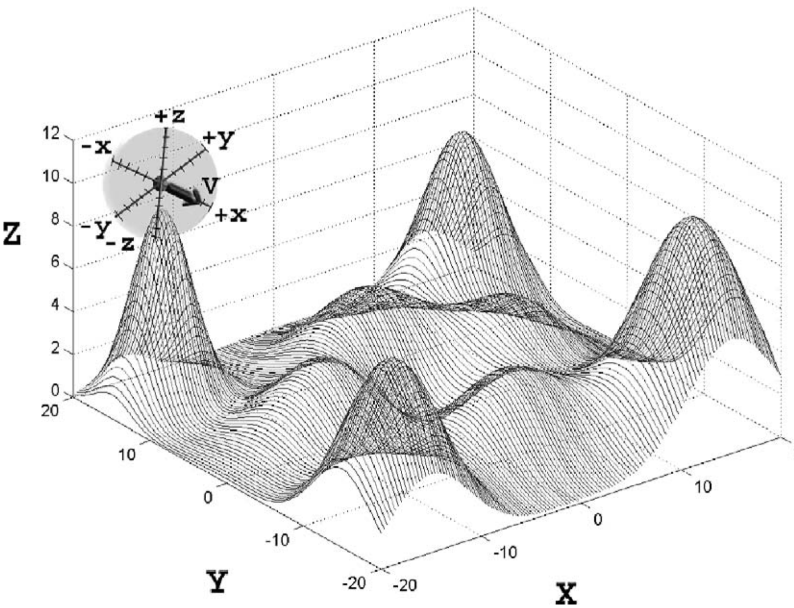


FIG. 2. An example of sampling the observation directions in two-dimensional problem.

a particle observing the landscape in six directions. The particle attempts to observe the search space for the landscape at several sampled distances from its position, in each direction. If the sampled point is within the landscape, the particle perceives the height of the landscape at that point. To be more realistic, the perception radius for observing the search space and other neighbouring particles can be separated into an inner radius and an outer radius. Within the inner perception radius, the particle has excellent perception, while its perception is less reliable in the outer perception range.

In the physical world, some social insects can perceive the presence of other individuals through other senses than those they use to observe the world. To simplify this in PPSO algorithm, particles can observe neighbouring particles in their perception range without sampling along specific directions. If there is any neighbour within the perception range, the particle perceives the approximate positions of neighbouring particles. The performance of each particle in the neighbourhood is unknown to each other. Therefore, each neighbouring particle might be in either a better or worse position than its own position. The particle chooses randomly the neighbouring particles, which will influence the particle to move towards them. The position of the chosen neighbour will be used as the local best position. If there is more than one neighbour chosen, the *lbest* position is the average position among those neighbours. The presence of the neighbouring particles influences the calculation of the new velocity for the next iteration in the same way as local social interaction, *lbest*, in the conventional particle swarm optimisation [28]. However, the particle will have no memory of the local best position from previous iterations. If the local best position at the current iteration does improve the performance of the particle, it will affect its personal best position in the next iteration because the *pbest* position is the position with maximum fitness value that the particle has ever been.

Apart from parameters in the conventional particle swarm optimisation, the main parameters of the perceptive particle swarm optimisation are: the perception radius, the number of observing directions and the number of points to observe along each observing direction. A larger perception radius allows more social interaction and encourages particles to explore the search space. This is because when there is no neighbouring particle within the perception range, the particle moves around its personal best position. However, the larger perception radius requires more computation time to observe the search space. A greater number of observing directions and a greater number of points to observe along each observing direction require more computation time as well. However, more observing directions allow a greater chance to obtain a good solution and the greater number of points offers more accuracy in observation. Note that the observation directions can be designed so that particles observe the search space at various angles in order to increase the chance that the swarm will find a good solution with acceptable computation time.

The PPSO algorithm is designed for optimisation problems in physical applications, such as a swarm of rescue robots searching for survivors after an earthquake, or micro or nanoscale robots used to construct a desired form, where the conventional PSO algorithm cannot be applied. In [30], an experimental validation was conducted in two-dimensional function optimisation problem. Despite the limited communication and performance measurements of the particles, the experiment showed comparable results with those from the conventional PSO algorithm [30].

6. Perceptive Particle Swarm Optimisation for Nanotechnology

Using this model of particle movement and perception, computers can be used to simulate the aggregation of various desired forms. While the PPSO algorithm can be used for function optimisation as described above, a more direct simulation enables the same algorithm to model bottom–up form generation. Instead of simulated particles randomly flocking in a virtual "function optimisation space," we can make them randomly flock in a virtual "form aggregation space." From the computer science

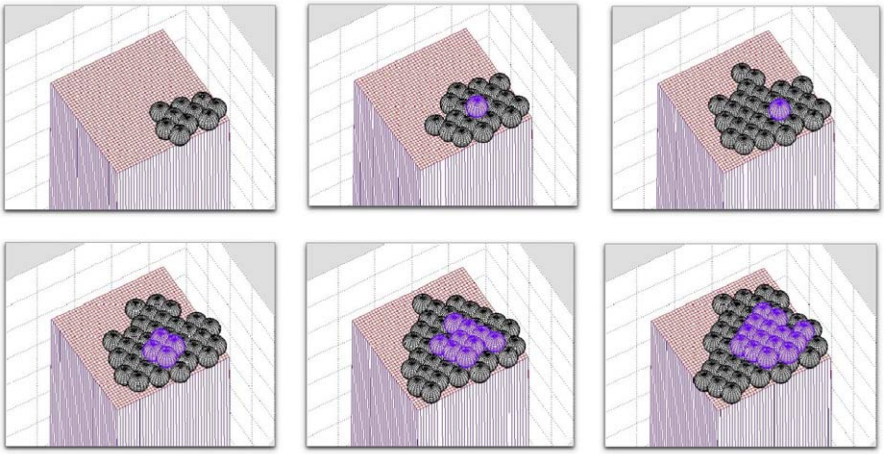


FIG. 3. A two-layered membrane is built from swarming nanoparticles over time.

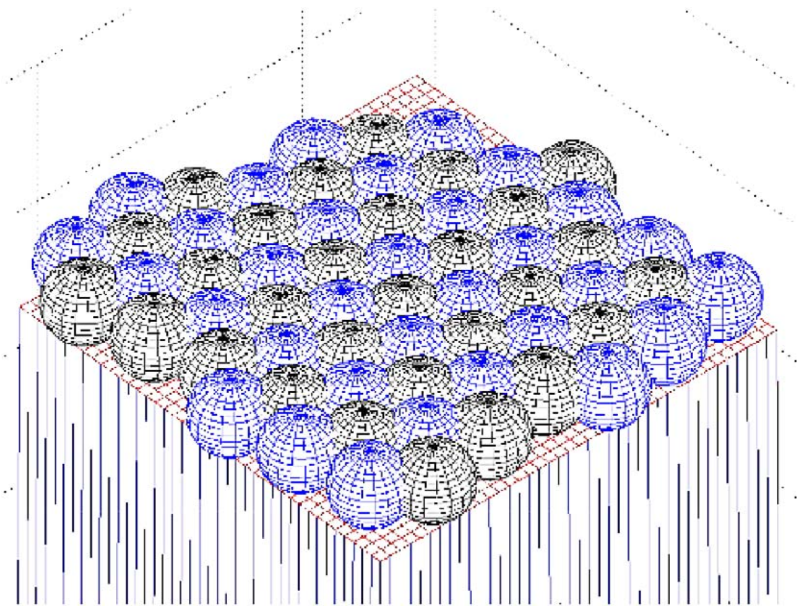


FIG. 4. A checkerboard pattern of two different types of particles.

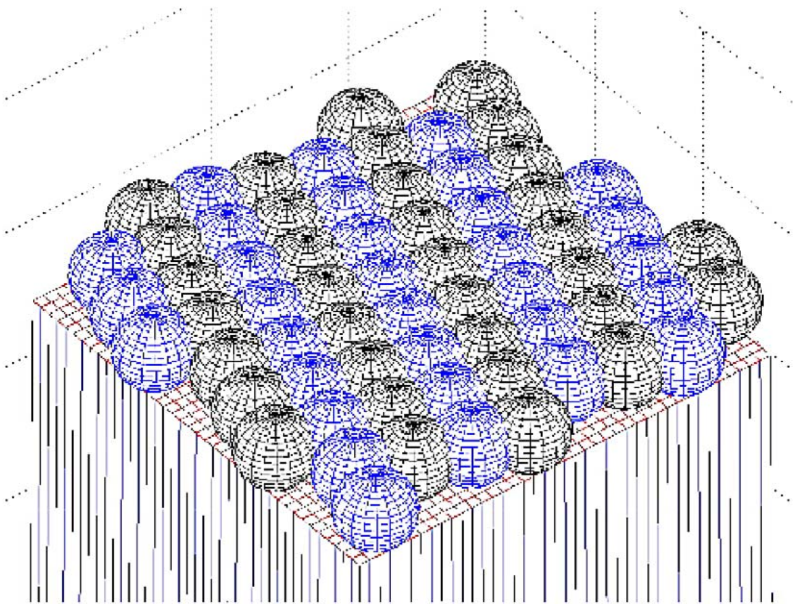


FIG. 5. A striped pattern of two different types of particles.

perspective, when using a swarm of particles to aggregate on a surface, the optimum becomes a series of optimal points (unlike typical optimisation problems). Using different rule sets for the swarming particles it is possible to model arbitrarily complex or simple attraction and repulsion behaviours.

By the addition of simple signals emitted by particles (which in a physical system might be equivalent to chemical gradients, electromagnetic fields or different adhesive properties) particles can be programmed to form in specific groups, patterns or layers. These larger structures emerge as particles are attracted to (and adhere to) a surface in the environment, and as they selectively adhere to each other. The movement of the particles is modelled as a free-floating cloud (i.e., akin to a gas or liquid) of flocking particles. Figure 3 illustrates how swarming particles slowly accumulate on a surface resulting in a two-layer membrane. Figures 4 and 5 show the results when the attraction ruleset is altered—checkerboard or striped patterns can be made to emerge.

The patterns shown in Figs. 4 and 5 are the result of applying constraints at the connectors of particles. For example, for the chess-board pattern all four connectors are to connect to the other type of particles. With such connection constraints, nanorobots just follow the attraction signal and then attach to the optimal ones with appropriate connector. No further consideration/computation is required.

7. Self-Assembling Nanotechnology

Swarm intelligence is not the only field that may inform future nanotechnology. The development of self-assembling robots has also taught us much.

Self-assembly (the autonomous construction of a device by itself) is a dream of robotics engineers and may be an essential requirement for future nanorobots. A payload of self-assembling components would be easier to transport to hazardous and distant locations compared to complete robots. A device that can self-assemble also has the ability to self-repair or regenerate damaged parts of itself, given replacement components. But the creation of self-assembling devices is a highly challenging problem.

The concept of self-assembling robots has been a popular theme in science fiction for many years. Only recently have robots been developed that display self-assembly characteristics. These robots are examples of netted systems [33], consisting of sensors and controllers that interact and self-assemble through data communication. These robots demonstrate the synthetic realisation of templated self-assembly [34], biological self-assembly [35], and self-reconfiguration [4,5], as examples from the disciplines of modular robotics and swarm robotics. However, such disciplines do not provide a generic methodology to creating self-assembling robots at all scales. This is largely due to scalability issues in relation to their respective methods of communication and assembly between modules or robotic-units.

Here, we refine the term self-assembly and suggest that it should be used to describe processes that can be controlled by an appropriate design of pre-existing components that interact in order to create emergent aggregate forms [33]. This view of self-assembly is used to link the principles self-assembly from nature to previous work in robotics and design.

Applying the principles of self-assembly to robotics has tremendous potential. This is especially true at the micro and nanoscale, where self-assembly is viewed as the only viable means of fabrication [33].

L.S. Penrose and R. Penrose were the first to show a mechanical analogue to natural self-assembly, specifically self-reproduction in the form of templated self-assembly [39]. They created two component shapes, labelled A and B, that connected in either an AB or BA configuration. Multiples of these A and B components were confined to a track in a random ordering, that when shaken, allowed components to move horizontally and interact with one another. By placing either an AB or a

BA seed complex on the track, it would cause neighbouring A and B or B and A components to self-assemble into AB and BA complexes respectively.

This example of templated self-assembly has recently been extended to robotics [34]. In this case, triangular-shaped programmed electromechanical components move randomly in two-dimensions on a cushion of air. When components collide, they communicate and latch and unlatch accordingly. Again, by initially placing a seed complex, free components can self-assemble and construct replicas of the seed complex [34].

In these two examples, templates are used to direct the self-assembly process of decentralised components. In contrast, swarm robotics uses swarm intelligence to direct the self-assembly process of decentralised robotic units, in a form of biological self-assembly. Of the robots produced in this discipline, Swarm-bot has shown successful results in mimicking self-assembling formations of social insects (e.g., the formation of living bridges by *Oecophylla longinoda* worker ants) [35]. Swarm-bot is the collective name to the set of cube-shaped mobile robotic units, named s-bots, which are capable of physically linking together. For example, s-bots can self-assemble into aggregate structures to move across terrain, otherwise not possible by an s-bot solely.

The discipline of modular robotics has produced self-reconfigurable robots using both centralised and decentralised modules [37]. Two of the most successful centralised modular robot implementations to date include PolyBot [36] and MTRAN [37]. These robots posses the ability to self-reconfigure a pre-existing set of modules that are physically connected together, and that move and attach/detach in terms of the degrees of freedom allowed by the components. PolyBot uses cube-shaped modules with one axis of rotation, which are capable of self-reconfiguring into various forms with movement such as in a loop, and in a snake-like and spider-like fashion [36]. MTRAN modules consist of two semi-cylindrical parts connected by a link, with each part being able to rotate 180° about its axis. These modules allow MTRAN to self-reconfigure into forms with one type of crawler and two types of quadruped movement [37].

These robots are all implementations of subsets of self-assembly, in the form of netted systems. In nature, self-assembly is primarily dictated by the design of the components within a system and the environmental conditions they are subjected to, as well as their component and environment physical and chemical properties [8,9]. The following section describes a general framework for self-assembling system, which covers the above mentioned types of self-assembly currently used in practise (templated self-assembly, biological self-assembly, and self-reconfiguration), and the potential to create self-assembling robots in the future, particularly at the nanoscale.

7.1 Framework

For the purposes of creating an artificial self-assembling system, the natural principles of self-assembly can be abstracted to four items:

- Components.
- Environment.
- Assembly protocol.
- Energy.

Components are defined by their properties. Such properties include, but are not limited to, shape, scale, material properties, and communication methods and interaction methods between components and/or their environment.

The environment in which components are subjected to can provide various functionalities, such as a boundary to which components are confined to. The physical and chemical properties of the environmental will also influence the nature in which components interact with one another, as well the way in which components self-assemble.

An assembly protocol defines the methods in which components can self-assemble (e.g., methods of attraction and repulsion). These methods are highly dependent on the scale of the system, as well as the physical and chemical properties of the components and the environment.

In order for the components to self-assemble, the components need to be mobile in their environment. This requires the components to have energy. This can either be available internally or transferred to components, for example, by the environment.

This self-assembly framework should be considered from the viewpoint of specific self-assembling systems. Physical constraints are normal in such systems, as we can observe in nature. A sand dune will only form in specific circumstances; if the wind force is not sufficient, it will not form. However, by continuing to gain a deeper understanding of self-assembly in nature, it can be leveraged for the purposes of design. This of course can be utilised by robotics, and the creation of simple self-assembling mechanical structures (e.g., pivots, joints, and levers), would be a fundamental next step.

7.2 Self-Assembly Illustration

One possible solution to creating simple structures is to utilise the relationship between component shape and an assembly protocol. Here, the relationship is investigated in the context of creating two-dimensional geometric mesoscale self-assembling structures, in a method that could be reduced to nanoscales.

Experiments were conducted to investigate whether a set of two-dimensional components (with concave and/or convex polygon shapes), could self-assemble into a desired shape. The assembly process is initiated by placing components on a tray, which is shaken in parallel to the surface of the tray. In this way energy is transferred to the components in the form of vibration, causing the components to move around and interact with one another; and magnetism is used to enable the components to attract and repel one another.

In this context, a component must have two essential properties; the first being the ability to fit together to form the desired shape and the second being the ability to join selectively to corresponding components or not to conflicting components. To achieve the first point, a set of components must include both concave and convex component shapes. By the components' shapes being both concave and convex, components are able to create stronger joints, leading to more stable structures overall, and less likely to break apart when colliding with other components or the sidewalls of the tray, compared to if components' shape were restricted to convex forms only. The second point in this example is achieved by placing a magnet in the interior of a non-magnetic material. The magnets allow components having opposite polarity to attract and assemble together, whereas components having similar polarity will repel each other, and therefore not assemble together. The non-magnetic material is used to determine the polygon form of the components. By not allowing the magnets in the components to join directly together, the components have a higher degree of freedom to move around in the given space and interact with one another. Figure 6 shows the principals behind the design of the components.

The components are placed on a tray, which allows a space in which the component shapes can move around and interact with one another. Movement of the components and their interaction is dictated by two-dimensional rigid body dynamics and magnetism. Figure 7 shows the three stable two-dimensional formations of magnetic discs.

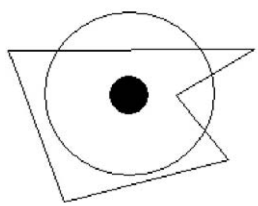


FIG. 6. Component design. The solid black circle represents the magnetic disc. The outer circle represents the area of the magnetic field. The irregular pentagon represents the non-magnetic material that defines the shape of the component (the left and bottom of the component are the areas not affected by the force of magnetism).

FIG. 7. The three stable two-dimensional formations of magnetic discs: grid (left); chain (centre); and triangular (right).

TABLE I
EXPERIMENTS

Experiment	Number of component shapes	Symmetric vs. non-symmetric component shapes	Magnetic formations
1. Triangle	4	Symmetric	Triangular
2. Square	4	Non-symmetric	Grid
3. Parallelogram	6	3 sets of 2 of symmetric shapes	Grid and chain
4. Irregular octagon	7	1 set of 4 symmetric shapes, and 3 non-symmetric shapes	Chain and triangular
5. 16-Sided polygon	10	Non-symmetric	Grid, chain, and triangular

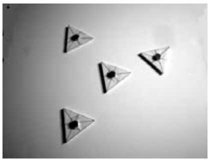
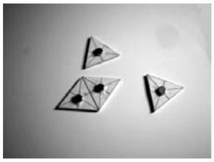
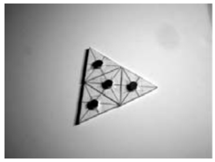
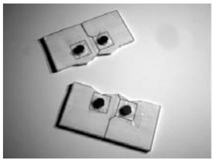
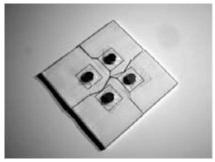
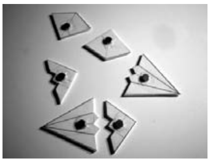
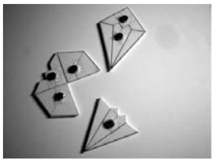
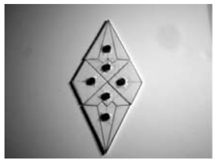
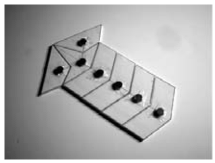
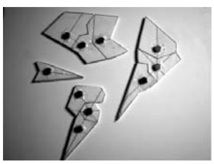
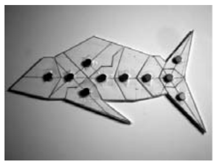
To test the validity of this design, five experiments were conducted. Components were constructed out of foam board, magnetic discs, and scotch tape. The tray was constructed out of foam board, pushpins, and general purpose adhesive. Each of the five experiments had a different number of components and different desired final forms. Symmetric and non-symmetric component shapes, along with the three stable two-dimensional magnetic disc formations, were also tested to see if they had an effect on the self-assembly process. Table I summarises the design and purposes of each of the five experiments.

7.2.1 Results

Each of the five experiments were successful in having their set of components self-assemble into their corresponding desired final form. Symmetric systems and systems with a lower number of components were able to self-assemble faster in general, in comparison to non-symmetric systems or systems with a large number of components. Table II shows the results of the five experiments.

These results demonstrate how the relationship between component shape and an assembly protocol can be used to create self-assembling entities of varying form. Although this combination of shape and an assembly protocol (magnetism), does not apply to all scales, it does however suggest that physical (as well as chemical), properties of a system can be leveraged to aide in creating netted systems.

TABLE II
EXPERIMENT RESULTS

Experiment	Configuration		
	Initial	Intermediate	Final
1. Triangle			
2. Square			
3. Parallelogram			
4. Irregular octagon			
5. 16-Sided polygon			

This relationship of component shape and an assembly protocol allowed for a larger set of feasible self-assembling entities (in the context of the experiment setup and design). In particular, this combination allowed for the exploitation of an effective magnetic force (regions of a component in which the effects of magnetism were subjected to neighbouring components), to create closed self-assembled forms. These forms, in contrast to open forms, do not allow for free components to self-assemble to the entity, when it reaches a particular state. Closed self-assembled forms are of particular interest to self-assembling robotics.

Another application of these results is that they demonstrate that physical properties, in this case shape and magnetic attraction/repulsion, can be used as a physical encoding, and as a communication mechanism between components. This concept could also be extended to chemical properties. Physical and chemical properties could be used to replace, simplify, or enhance communication and interaction mechanisms between modules or robotic-units, in self-assembling robots.

Understanding and utilising the principles of self-assembly in nature, could be used for the realisation of nanorobots. At the nanoscale, self-assembly is considered as the only viable means of fabricating entities [33]. At this scale, as well as all others, numerous variables affect the process of self-assembly. Optimisation algorithms can be used to generate the specifications of the components and environment to create self-assembling systems [38].

7.3 Evolutionary Computation Model

With its ability to navigate through complex problem spaces, evolutionary computation has proven to be an extremely useful approach to solving optimisation problems. In addition, evolutionary computation can be used as a creative tool. This is most notably seen in its ability to generate novel designs [42]. This duality of evolutionary computation makes it a prime candidate to be incorporated into a process for designing and physically creating self-assembling systems [38].

One embodiment of the framework described earlier (and in [38]) can be described as an eight-step process. These steps include:

1. Define the properties of the desired self-assembling entity.
2. Encapsulate the component design, environment design, and/or construction process (referring to the methodology in which the components and/or environment would physically be created, e.g., using rapid prototyping techniques). This encapsulated information would be encoded into the genotype and phenotype representations of the components and environment.
3. Define the translation process in which the computer generated designs can be used to physically create the components and/or environment (e.g., translating the software representations of the components and/or environment to CAD files).
4. Create software that incorporates a computer model using evolutionary computation to virtually design and test the candidate components and/or environment, to allow for self-assembly of the components into the desired entity.
5. Execute the software to generate the designs of the components and/or environment.
6. Execute the translation process of the computer generated component/and or environment designs, to a form that can be used for physical fabrication.

7. Build the components and/or environment.
8. Place the components in their environment to allow for the components to self-assemble into the desired entity.

In this evolutionary computation model, the notion of a design space (the set of buildable designs of components and/or environment), is crucial. If this space is ill-defined, it will greatly affect the performance of the software, as well as inhibit the creation of the self-assembling system.

The encapsulation of a design space is a complicated task. However, it is of great importance, especially in using this process [38] for creating physical system, such as self-assembling robots. Using this process [38] to create simple self-assembling entities with features of simple mechanical machines (e.g., pivots, joints, and levers), would be an important next step, with benefits from the macroscale to the nanoscale. Preliminary results from an implementation of this embodiment, which used a genetic algorithm to evolve shapes and a simulator to model their interaction are described in [43]. Work on this area is ongoing by the authors.

8. Conclusions

As the development of nanotechnology progresses in several disciplines including physics, chemistry, biology and material science, computer scientists must be aware of their roles and brace themselves for the greater advancement of nanotechnology in the future. This chapter has outlined the development of nanotechnology. It is hoped that this gentle review will benefit computer scientists who are keen to contribute their works to the field of nanotechnology. We also suggested the possible methods that computer science can offer in the task of programming future nanotech, which can benefit other nanotechnologists from other fields by helping them be aware of the opportunities from computer science.

As computer scientists who are interested in the field of nanotechnology, our current work involves building systems that consist of a large number of particles automatically forming into a designed structure. By using the PPSO algorithm to control the swarm of particles, each particle performs lightweight computations and holds only a few values. It is anticipated that models such as these will lead to successful bottom–up nanotechnology systems in the future.

In addition, the principles of self-assembly in nature should be considered when creating self-assembling devices. The general framework presented here (consisting of a set of components, an environment, an assembly protocol, and energy), provides a method for understanding the requirements of a specific self-assembling system.

There can be no doubt that nanotechnology will play a major role in our future technology. In this chapter we have outlined some of the ways in which computer science is assisting this research effort.

Uncited references

[40] [41]

References

[1] Drexler K.E., *Engines of Creation: The Coming Era of Nanotechnology*, Anchor Press, 1986.

[2] Drexler K.E., Peterson C., Pergamit G., "Unbounding the future: The nanotechnology revolution", 1991.

[3] Whitesides G.M., Love J.C., *The Art of Building Small*, Scientific American, September 2001.

[4] Roukes M.L., *Plenty of Room, Indeed*, Scientific American, September 2001.

[5] Levit C., Bryson S.T., Henze C.E., "Virtual mechanosynthesis", in: *Fifth Foresight Conference on Molecular Nanotechnology*, 1997.

[6] Falvo M.R., Clary G.J., Taylor R.M., Chi V., Brooks F.P., Washburn S., Superfine R., "Bending and buckling of carbon nanotubes under large strain", *Nature* **389** (1997) 582–584.

[7] Hall J.S., "Agoric/genetic methods in stochastic design", in: *Fifth Foresight Conference on Molecular Nanotechnology*, 1997.

[8] Drexler K.E., *Machine-Phase Nanotechnology*, Scientific American, September 2001.

[9] Stix G., *Little Big Science*, Scientific American, September 2001.

[10] Eigler D.M., Schweizer E.K., "Positioning single atoms with a scanning tunneling microscope", *Nature* **344** (1990) 524.

[11] Taylor R.M. II, Robinett W., Chi V.L., Brooks F.P., Wright W.V. Jr., Williams R.S., Snyder E.J., "The nanomanipulator: A virtual-reality interface for a scanning tunneling microscope", in: *Proceedings of SIGGRAPH*, 1993.

[12] Rai-Choudhury P. (Ed.), *Handbook of Microlithography, Micromachining and Microfabrication*, 1997.

[13] Muller W.T., Klein D.L., Lee T., Clarke J., McEuen P.L., Schultz P.G., "A strategy for the chemical synthesis of nanostructures", *Science* **268** (1995) 272–273.

[14] Snow E.S., Campbell P.M., "AFM fabrication of sub-10-nanometer metal-oxide devices with in situ control of electrical properties", *Science* **270** (1995) 1639–1641.

[15] Rosei F., Schunack M., Jiang P., Gourdon A., Laegsgaard E., Stensgaard I., Joachim C., Besenbacher F., "Organic molecules acting as templates on metal surfaces", *Science* **296** (2002) 328–331.

[16] Hla S.W., Rieder K.H., "STM control of chemical reactions: Single-molecule synthesis", *Annu. Rev. Phys. Chem.* **54** (2003) 307–330.

[17] Iijima S., "Helical microtubules of graphitic carbon", *Nature* **354** (1991).

[18] Fulton T.A., Dolan G.J., "Observation of single-electron charging effects in small tunnel junctions", *Phys. Rev. Lett.* **59** (1987) 109–112.

[19] Lieber C.M., *The Incredible Shrinking Circuit*, Scientific American, September 2001.

[20] Reed M.A., Tour J.M., *Computing with Molecules*, Scientific American, June 2000.

[21] Adleman L., "Molecular computation of solutions to combinatorial problems", *Science* **266** (1994) 1021–1024.

[22] Gershenfeld N., *Quantum Computing with Molecules*, Scientific American, June 1998.

[23] Feynman R., "Simulating physics with computers", *Internat. J. Theoret. Phys.* **21** (1982) 467–488.

[24] Drexler K.E., *Nanosystems: Molecular Machinery, Manufacturing, and Computation*, Wiley, 1992.

[25] Astumian R.D., *Making Molecules into Motors*, Scientific American, July 2001.

[26] Ashley S., *Nanobot Construction Crews*, Scientific American, September 2001.

[27] Eberhart R., Shi Y., Kennedy J., *Swarm Intelligence*, Morgan Kaufmann, 2001.

[28] Bonabeau B., Dorigo M., Theraulaz G., *Swarm Intelligence: From Natural to Aritificial Systems*, Oxford University Press, 1999.

[29] Holland O.E., Melhuish C.R., "Getting the most from the least: Lessons for the nanoscale from minimal mobile agents", in: *Aritificial Life V*, 1996.

[30] Kaewkamnerdpong B., Bentley P.J., "Perceptive particle swarm optimisation", in: *Proceedings of International Conference on Adaptive and Natural Computing Algorithms, ICANNGA*, 2005.

[31] Kumar S., Bentley P.J. (Contributing Eds.), *On Growth, Form and Computers*, Academic Press, London, 2003.

[32] Bonsma E., Karunatillake N.C., Shipman R., Shackleton M., Mortimore D., "Evolving greenfield passive optical networks", *BT Technol. J.* **21** (4) (2003) 44–49.

[33] Whitesides G.M., Grzybowski B., "Self-assembly at all scales", *Science* **295** (5564) (2002) 2418–2421.

[34] Grifth S., Goldwater D., Jacobson J., "Robotics: Self-replication from random parts", *Nature* **437** (7059) (2005) 636.

[35] Gross R., Dorigo M., Yamakita M., "Self-assembly of mobile robots: From swarm-bot to super-mechano colony", in: *Proc. of the 9th Int. Conf. on Intelligent Autonomous Systems, IAS-9*, 2006, pp. 487–496.

[36] Duff D.G., Yim M., Roufas K., "Evolution of PolyBot: A modular reconfigurable robot", in: *Harmonic Drive International Symposium*, 2001.

[37] Kamimura A., Murata S., Yoshida E., Kurokawa H., Tomita K., Kokaji S., "Self-reconfigurable modular robot—Experiments on reconfiguration and locomotion", in: *Proc. of the 2001 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS '01*, 2001, pp. 606–612.

[38] Bhalla N., Bentley P.J., "Self-assembling systems", UK Patent 0614400.0, July 19, 2006.

[39] Penrose L.S., Penrose R., "A self-reproducing analogue", *Nature* **179** (4571) (1957) 1183.

[40] Ball P., *The Self-Made Tapestry: Pattern Formation in Nature*, Oxford University Press, 1999.
[41] Thompson D.W., *On Growth and Form*, Cambridge University Press, 1917.
[42] Bentley P.J., "The revolution of evolution in design: From coffee tables to hospitals", in: *Proc. of Recent Advances in Soft Computing '98*, 1998, pp. 172–182.
[43] Bhalla N., "Self-assembling systems in 2 dimensions", MSc intelligent systems project dissertation, Department of Computer Science, University College London, 2004.
[44] Reed M.A., Tour J.M., *Computing with Molecules*, Scientific American, June 2000.
[45] Holland O., Melhuish C., "Stigmergy, self-organisation, and sorting in collective robotics", *Artif. Life* **5** (2) (1999) 173–202.
[46] Deneubourg J.-L., Goss S., Franks N., Sendova-Franks A., Detrain C., Chretien L., "The dynamics of collective sorting: Robot-like ant and ant-like robot", in: Mayer J.A., Wilson S.W. (Eds.), *Simulation of Adaptive Behavior: From Animals to Animats*, MIT Press, 1991, pp. 356–365.
[47] Beckers R., Holland O.E., Deneubourg J.-L., "From local actions to global tasks: Stigmergy and collective robotics", in: *Proceedings of the Fourth Workshop on Artificial Life*, MIT Press, 1994, pp. 181–189.
[48] Bonabeau E., Theraulaz G., Deneubourg J.-L., Aron S., Camazine S., "Selforganization in social insects", *Trends Ecol. Evol.* **12** (1997) 188–193.
[49] Beni G., Wang J., "Swarm intelligence", in: *Proceedings of the Seventh Annual Meeting of the Robotics Society of Japan*, RSJ Press, 1989, pp. 425–428.
[50] Beni G., Wang J., "Swarm intelligence in cellular robotic systems", in: *Proceedings of NATO Advanced Workshop on Robotics and Biological Systems*, 1989.
[51] Colorni A., Dorigo M., Maniezzo V., "Distributed optimization by ant colonies", in: *Proceedings of the First European Conference on Artificial Life*, 1991, pp. 134–142.
[52] Dorigo M., Di Caro G., "The ant colony optimization meta-heuristic", in: Corne D., Dorigo M., Glover F. (Eds.), *New Ideas in Optimization*, McGraw-Hill, 1999, pp. 11–32.
[53] Deneubourg J.-L., Goss S., Pasteels J.M., Fresneau D., Lachaud J.-P., "Selforganization mechanisms in ant societies (II): Learning in foraging and division of labour", *Experientia* **54** (Suppl.) (1987) 177–196.
[54] Dorigo M., Gambardella L.M., "Ant colonies for the traveling salesman problem", *BioSystems* **43** (1997) 73–81.
[55] Emlen J.T. Jr., "Flocking behavior in birds", *Auk* **69** (2) (1952) 160–170.
[56] Heppner F., "Three-dimensional structure and dynamics of bird flocks", in: Parrish J.K., Hammer W.M. (Eds.), *Animal Groups in Three Dimensions*, Cambridge University Press, 1997, pp. 68–89.
[57] Kennedy J., Eberhart R., "Particle swarm optimization", in: *Proceedings of the IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.
[58] Resnick M., *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*, MIT Press, 1997.
[59] Reynolds C.W., "Flocks, herds, and schools: A distributed behavioural model", *Computer Graphics* **21** (4) (1987) 25–34.

[60] Whitesides G.M., Love J.C., *The Art of Building Small*, Scientific American, September 2001.

[61] Fukuyama Y., Yoshida H., "A particle swarm optimization for reactive power and voltage control in electric power systems", in: *Proceedings of the IEEE Congress on Evolutionary Computation*, 2001.

[62] Blackwell T., Bentley P.J., "Improvised music with swarms", in: *Proceedings of the IEEE Congress on Evolutionary Computation*, 2002.

[63] Omran M., Salman A., Engelbrecht A.P., "Image classification using particle swarm optimization", in: *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, 2002*, 2002, pp. 370–374.

[64] Coello C.A., Luna E.H., Aguirre A.H., "Use of particle swarm optimization to design combinational logic circuits", in: *Lecture Notes in Comput. Sci.*, vol. 2606, 2003, pp. 398–409.

[65] Ujjin S., Bentley P.J., "Particle swarm optimization recommender system", in: *Proceedings of the IEEE Swarm Intelligence Symposium*, 2003, pp. 124–131.

[66] Salerno J., "Using the particle swarm optimization technique to train a recurrent neural model", in: *Proceedings of IEEE International Conference on Tools with Artificial Intelligence*, 1997, pp. 45–49.