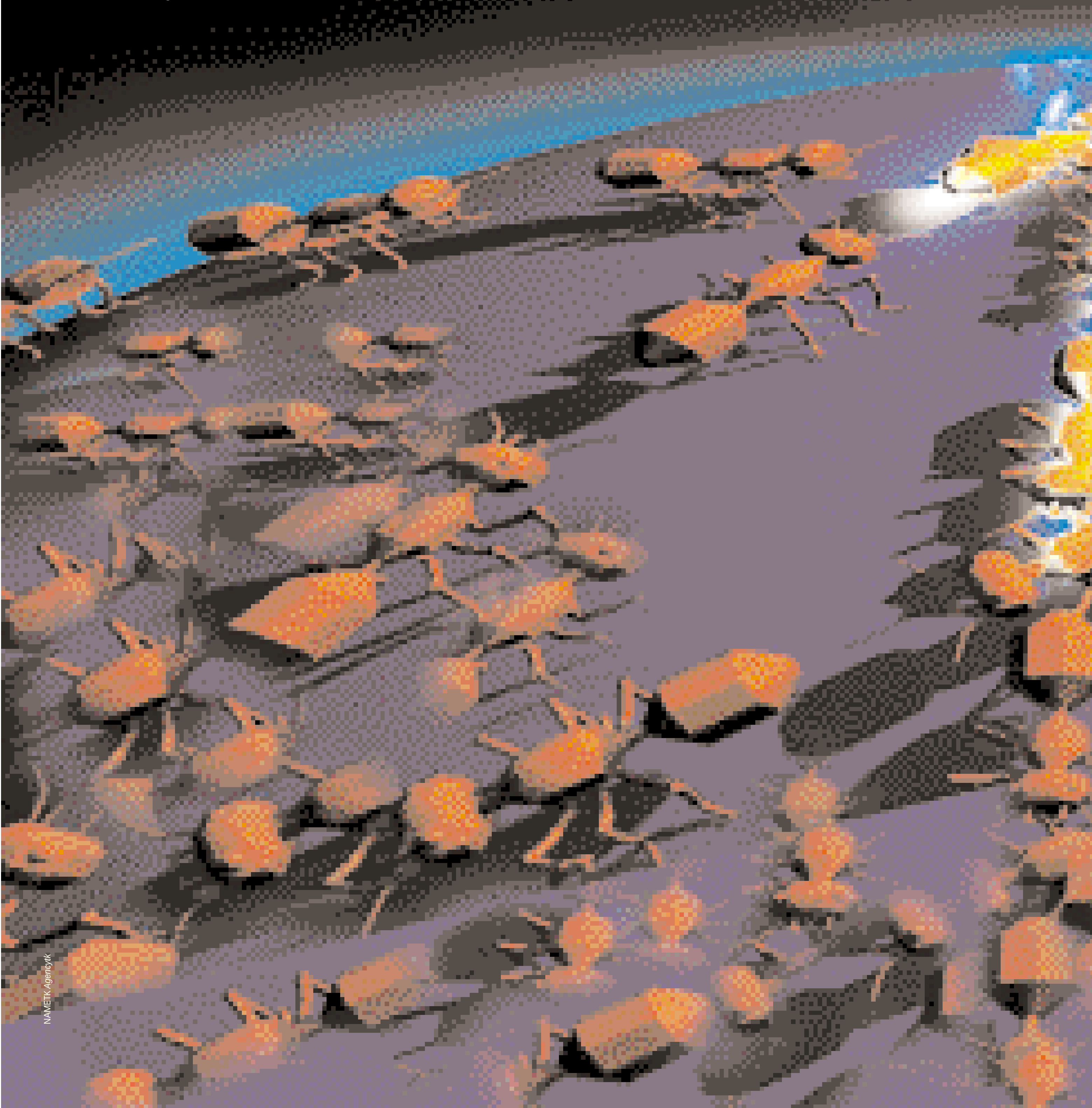
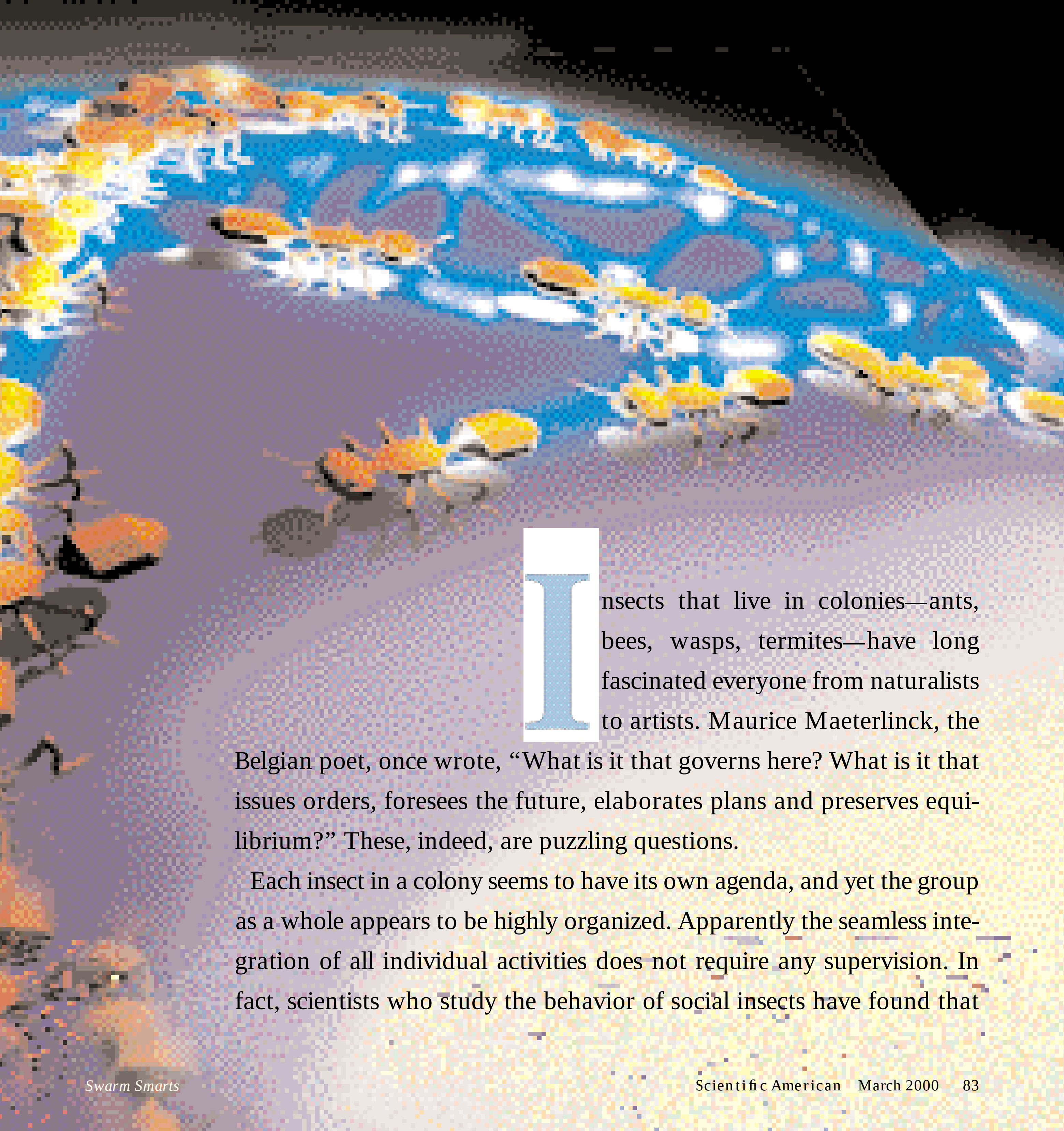
SWARM SMARTS

by Eric Bonabeau and Guy Théraulaz



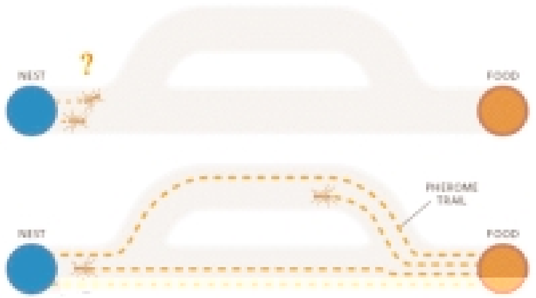
NAMETK *Agencytk*

Using ants and other social insects as models, computer scientists have created software agents that cooperate to solve complex problems, such as the rerouting of traffic in a busy telecom network



Insects that live in colonies—ants, bees, wasps, termites—have long fascinated everyone from naturalists to artists. Maurice Maeterlinck, the Belgian poet, once wrote, “What is it that governs here? What is it that issues orders, foresees the future, elaborates plans and preserves equilibrium?” These, indeed, are puzzling questions.

Each insect in a colony seems to have its own agenda, and yet the group as a whole appears to be highly organized. Apparently the seamless integration of all individual activities does not require any supervision. In fact, scientists who study the behavior of social insects have found that



NAMETK *Agencytk*

PHEROMONE TRAILS enable ants to forage efficiently. Two ants leave the nest at the same time (*top*), each taking a different path and marking it with pheromone. The ant that took the shorter path returns first (*bottom*). Because this trail is now doubly marked with pheromone, it will attract other ants more than the longer route will.

DIFFERENT FOOD SOURCES are raided sequentially because of pheromone evaporation. In this computer simulation, three identical sources of food are located at unequal distances from a nest (*a*). After foraging randomly (*b*), the ants begin to raid the food that is closest (*c*). As those supplies dwindle (*d*), the concentration of pheromone along their trails decreases through evaporation. The ants will then exploit the farther source.

NAMETK *Agencytk*

NETWORK TRAFFIC can be rerouted on-the-fly with software agents that mimic ants. A transmission that needs to travel from A to B must go through a number of intermediate nodes. If the shortest path (*red*) between the two locations is congested, the system must redirect the transmission through an alternative (*blue*). Software agents can perform this rerouting automatically in a manner that is similar to how ants raid different food sources (*illustration above*). In the analogy, a congested path is like a depleted food source.

NAMETK *Agencytk*

cooperation at the colony level is largelyself-organized: in numerous situations the coordination arises from interactions among individuals. Although these interactions might be simple (one ant merely following the trail left by another), together they can solve difficult problems(finding the shortest route among countless possible paths to a food source). This collective behavior that emerges from a group of social insects has been dubbed “swarm intelligence.”

Recently a growing community of researchers has been devising new ways of applying swarm intelligence to diverse tasks. The foraging of ants has led to a novel method for rerouting network traffic in busy telecommunications systems. The cooperative interaction of ants working to transport a large food item may lead to more effective algorithms for robots. The way in which insects cluster their colony’s dead and sort their larvae can aid in analyzing banking data. And the division of labor among honeybees could help streamline the assembly lines in factories.

Virtual Foraging

One of the early studies of swarm intelligence investigated the foraging behavior of ants. Jean-Louis Deneubourg of the Free University of Brussels and his colleagues showed that the ant “highways” often seen in nature (and in people’s kitchens) result from individual ants exuding pheromone, a chemical substance, that attracts other ants. Deneubourg, a pioneer in the field, also demonstrated that this process of laying a trail of pheromone that others can follow was a good strategy for finding the shortest path between a nest and a food source.

In experiments with the Argentine ant *Linepithema humile,* Deneubourg constructed a bridge with two branches, one twice as long as the other, that separated a nest from a food source. Within just a few minutes the colony usually selected the shorter branch. Deneubourg found that the ants lay and follow trails as they forage. Individual ants expel pheromone, which attracts other ants. The first ants returning to the nest from the food source are those that have taken the shorter path in both directions, from the nest to the food and back. Because this route is the first to be doubly marked with pheromone, nestmates are attracted to it.

If, however, the shorter branch is presented to the colony after the longer branch, the ants will not take it because the longer branch has already been marked with pheromone. But computer scientists can overcome this problem in an artificial system by introducing pheromone decay: when the chemical evaporates quickly, longer paths will have trouble maintaining stable pheromone trails. The software ants can then select a shorter branch even if it is discovered belatedly. This property is highly desirable to prevent the system from converging on mediocre solutions. (In *L. humile,* the pheromone concentrations do decay but at a very slow rate.)

In a computer simulation of pheromone evaporation [*see illustration on page 00*], researchers presented identical food sources to an artificial colony at different distances from the nest. At first the virtual ants explored their environment randomly. Then they established trails that connected all of the food sources to the nest. Next they maintained only the trails of the sources closest to the nest, leading to the exploitation of those supplies. With the depletion of that food, the software ants began to raid the farther sources.

Extending this ant model, Marco Dorigo, a computer scientist at the Université Libre de Bruxelles, and his colleagues

In the traveling salesman problem, a person must find the shortest route by which to visit a given number of cities, each exactly once. The classic problem is devilishly difficult: for just 15 cities [*see illustration at right*] there are billions of route possibilities.

Recently researchers have begun to experiment with antlike agents to derive a solution. The approach relies on the artificial ants laying and following the equivalent of pheromone trails [*see illustrations on opposite page*].

Envision a colony of such ants, each independently hopping from city to city, favoring nearby locations but otherwise traveling randomly. After completing a tour of all the cities, an ant goes back to the links it used and deposits pheromone. The amount of the chemical is inversely proportional to the overall length of the tour: the shorter the distance, the more pheromone each of the links receives. Thus, after all the ants have completed their tours and spread their pheromone, the links that belonged to the highest number of short tours will be richest with the chemical. Because the pheromone evaporates, links in long routes will eventually contain significantly less of the substance than those in short tours.

The colony of artificial ants is then released to travel over the cities again, but this time they are guided by the earlier pheromone trails (high-concentration links are favored) as well as by the intercity distances (nearby locations have priority), which the ants can obtain by consulting a table storing those numbers. In general, the two criteria—pheromone strength and intercity distance—are roughly weighted equally.

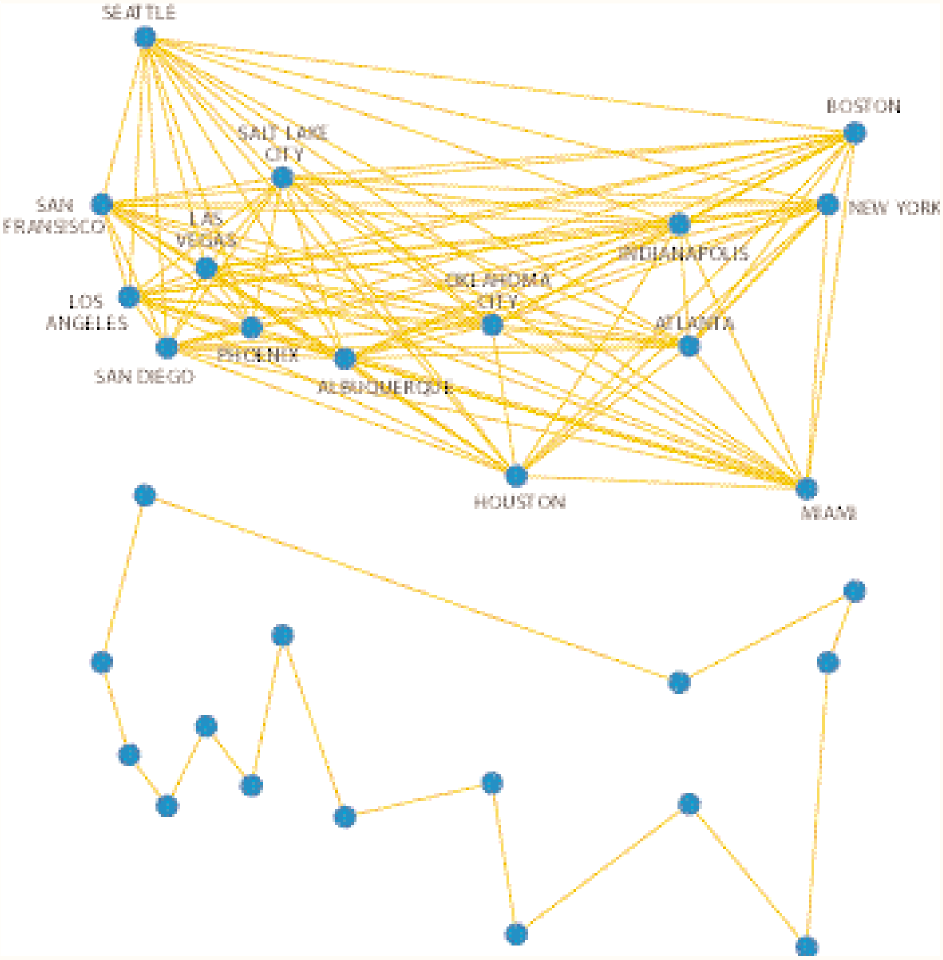
Marco Dorigo of the Free University of Brussels and his colleagues have implemented this ant-based system in software. Of course, the methodology assumes that the favored links, when taken together, will lead to an overall short route. Dorigo has found that after repeating the process (tour completion followed by pheromone reinforcement and evaporation) numerous times, the artificial ants are indeed able to obtain progressively shorter tours, such as that shown at the right.

Nevertheless, a difficulty arises when many routes happen to use a link that, as it turns out, is not part of a short tour. (In fact, such a link might belong to many, many long routes.) Dorigo discovered that although this popular link might bias the search for several iterations, a better connection will eventually replace it. This optimization is a consequence of the subtle interplay between reinforcement and evaporation, which ensures that only the better links survive. Specifically, at some point an alternative connection that is part of a short route would be selected by chance and would become reinforced more than the popular link, which would then lose its attractiveness as its pheromone evaporated.

Another problem occurs when a short route contains a very long link that initially is less likely to be used. But Dorigo has shown that even though the connection might be a slow starter, once it has been selected it will quickly become reinforced more than other, competing links.

It is important to note that this ant-based method is effective for finding short routes but not necessarily the shortest one. Nevertheless, such near-optimal solutions are often more than adequate, particularly because obtaining the best route can require an unwieldy amount of computation. In fact, determining the exact solution quickly becomes intractable as the number of cities increases.

In addition, Dorigo's system has one advantage: its inherent flexibility. Because the artificial ants are continuously exploring different paths, the pheromone trails provide backup plans. So, whenever one of the links breaks down (bad weather between Houston and Atlanta, for instance), a pool of alternatives already exists.



have devised a way to solve the famous "traveling salesman problem" [*see box on page 00*]. The problem calls for finding the shortest route that goes through a given number of cities exactly once. This test is appealing because it is easy to formulate and yet extremely difficult to solve. It is "NP-complete": the solution requires a number of computational steps that grows faster than the number of cities raised to any finite power (NP stands for nondeterministic polynomial). For such problems, people usually try to find an answer that is good enough but not necessarily the best (that is, a route that is sufficiently short but perhaps not the shortest). Dorigo has shown that he can obtain near-optimal routes by using artificial ants that are tweaked so that the concentration of pheromone they deposit varies with the overall distances they have traveled.

Similar approaches have been successful in a number of other optimization tasks. For instance, artificial ants provide the best solution to the classic quadratic assignment problem, in which the manufacture of a number of goods must be assigned to different factories so as to minimize the total distance over which the items need to be transported between facilities. In a related application, Dave Gregg of Unilever in the U.K. and Vince Darley of Bios Group in Santa Fe, N.M., report that they have developed an ant-based method for decreasing the time it takes to perform a given amount of work in a large Unilever plant. The system must efficiently schedule various storage tanks, chemical mixers, packaging assembly lines and other equipment.

Dynamic Ants

In addition to solving optimization problems that are basically static, or nonvarying, antlike agents can also cope with glitches and dynamic environments—for example, a factory where a machine breaks down. By maintaining pheromone trails and continuously exploring new paths, the ants serendipitously set up a backup plan and thus are prepared to respond to changes in their environment. This property, which may explain the ecological success of real ants, is crucial for many applications.

Consider the dynamic unpredictability of a telephone network. A phone call from A to B generally has to go through a number of intermediate nodes, or switching stations, requiring a mechanism to tell the call where it should hop next to establish the A-to-B connection. Obviously the algorithm for this process should avoid congested areas to minimize delays, and backup routes become especially valuable when conditions change dramatically. Bad weather at an airport or a phone-in competition on TV will lead to transient local surges of network traffic, requiring on-the-fly rerouting of calls through less busy parts of the system.

To handle such conditions, Ruud Schoonderwoerd and Janet Bruten of Hewlett-Packard's research laboratories in Bristol, U.K., and Owen Holland of the University of the West of England have invented a routing technique in which antlike agents deposit bits of information, or "virtual pheromone," at the network nodes to reinforce paths through uncongested areas. Meanwhile an evaporation mechanism adjusts the node information to disfavor paths that go through busy areas.

Specifically, each node keeps a routing table that tells phone

Cooperative Transport in Ants and Robots


NAMETK Agencytk

In some ant species, nestmates are recruited to help when a single ant cannot retrieve a large prey. Then, during an initial period that can last up to several minutes, the ants change their positions and alignments around the object until they are able to move the prey toward their nest.

Using mechanical robots, Ron Kube and Hong Zhang of the University of Alberta have reproduced this behavior. The task for their robotic army was to push a box toward a goal, and each individual was programmed with very simple instructions: find

ANTS WORK TOGETHER to transport a large leaf (*left*). Such teamwork has inspired scientists to program robots without the use of complex software. In an experiment at the University of Alberta (*below*), the robots must push an illuminated circular box toward a light. Even though each robot (*right*) does not communicate with the others and acts independently by following a small set of simple instructions, together the group is able to accomplish its goal.

calls where to go next depending on their destinations. Antlike agents continually adjust the table entries, or scores, to reflect the current network conditions. If an agent experiences a long delay because it went through a highly congested portion of the network, it will add just a tiny amount of "pheromone" to the table entries that would send calls to that overloaded area. In mathematical terms, the scores for the corresponding nodes would be increased just slightly. On the other hand, if the agent went quickly from one node to another, it would reinforce the use of that path by leaving a lot of "pheromone"—that is, by increasing the appropriate scores substantially. The calculations are such that even though a busy path may by definition have many agents traveling on it, their cumulative "pheromone" will be less than that of an uncongested path with fewer agents.

The system removes obsolete solutions by applying a mathematical form of evaporation: all of the table entries are decreased regularly by a small amount. This process and the way in which the antlike agents increase the scores are designed to work in tandem so that busy routes experience more evaporation than reinforcement, whereas uncongested routes undergo just the opposite.

Any balance between evaporation and reinforcement can be disrupted easily. When a previously good route becomes congested, agents that follow it are delayed, and evaporation overcomes reinforcement. Soon the route is abandoned, and the agents discover (or rediscover) alternatives and exploit them. The benefits are twofold: when phone calls are rerouted through the better parts of a network, the process not only allows the calls to get through expeditiously but also enables the congested areas to recover from the overload.

Several companies are exploring this approach for handling the traffic on their networks. France Télécom and British Telecom have taken an early lead in applying ant-based routing methods to their systems. In the U.S., MCI-Worldcom has been investigating artificial ants not only for managing the company's telephone network but also for other tasks such as customer billing. The ultimate application, though, may be on the Internet, where traffic is particularly unpredictable.

To handle the demanding conditions of the Net, Dorigo and his colleague Gianni Di Caro of the Université Libre de Bruxelles have increased the sophistication of the ant agents by taking into account several other factors, including the overall time it takes information to get from its origin to its destination. (The approach for phone networks considers just the time it takes to go from one node to another, and the traffic in the reverse direction is assumed to be the same.) Simulation results indicate that Dorigo and Di Caro's system outperforms all other routing methods in terms of both maximizing throughput and minimizing delays. In fact, extensive tests suggest that the ant-based method is superior to Open Shortest Path First, the protocol that the Internet currently uses, in which nodes must continually inform one another of the status of the links to which they are connected.

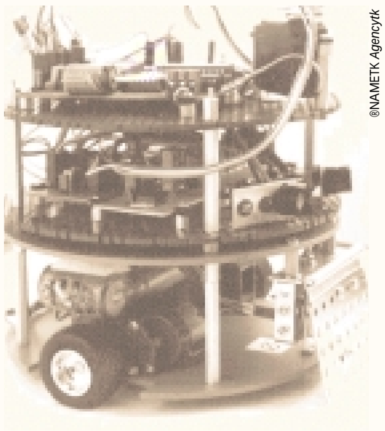
A Swarm of Applications

Other behaviors of social insects have inspired a variety of research efforts. Computer scientists are studying insect swarms to devise different techniques for controlling a group of robots. One application being investigated is cooperative transport [*see box on page 00*]. Using such approaches, engineers could design relatively simple and cheap robots that would work together to perform increasingly sophisticated tasks. In

the box, make contact with it, position yourself so that the box is between you and the goal, then push the box toward the goal.

Although the robots were intentionally programmed very crudely, the similarity between their behavior and that of a swarm of ants is striking. (The videotaped experiments can be viewed at http://www.cs.ualberta.ca/~kube/ on the World Wide Web.) At first, the robots move randomly, trying to find the box. After locating it they begin pushing, but if they are unsuccessful in moving it they change their positions and alignments. Even temporary setbacks are evident, as when the box is moved in a direction away from the goal. The robots make continual adjustments when they lose contact with the box, when they block one another or when the box rotates. Eventually the robots, despite their limited capabilities, are successful in delivering the box to the goal.

Obviously, individuals trying to push an object can find far more efficient ways to work together. But because of the extreme simplicity of this ant-based approach—for one thing, the robots do not need to communicate with one another—it is promising for miniaturization and low-cost applications.



©NAMETK *Agencytk*

NAMETK *Agencytk*

From Cemeteries to Databases

In some ant species, such as *Messor sancta,* workers pile up their colony's dead to clean their nests. The illustration at the right shows the dynamics of such cemetery organization. If the corpses are randomly distributed at the beginning of the experiment, the workers will form clusters within a few hours.

Jean-Louis Deneubourg of the Free University of Brussels and his colleagues have proposed a simple explanation: small groups of items grow by attracting workers to deposit more items, and this positive feedback leads to the formation of larger and larger bunches. Scientists, however, still do not know the exact details of the individual behavior that implements the feedback mechanism.

Another phenomenon can be explained in a similar way. The workers of the ant *Leptothorax unifasciatus* sort the colony's brood systematically. Eggs and microlarvae are placed at the center of an area, the largest larvae at the periphery, and pupae and prepupae in between. One explanation of this behavior is that ants pick up and drop items according to the number of similar surrounding objects. For example, if an ant finds a large larva surrounded by eggs, it will most likely pick up the larval "misfit." And that ant will probably deposit its load in a region containing other large larvae.

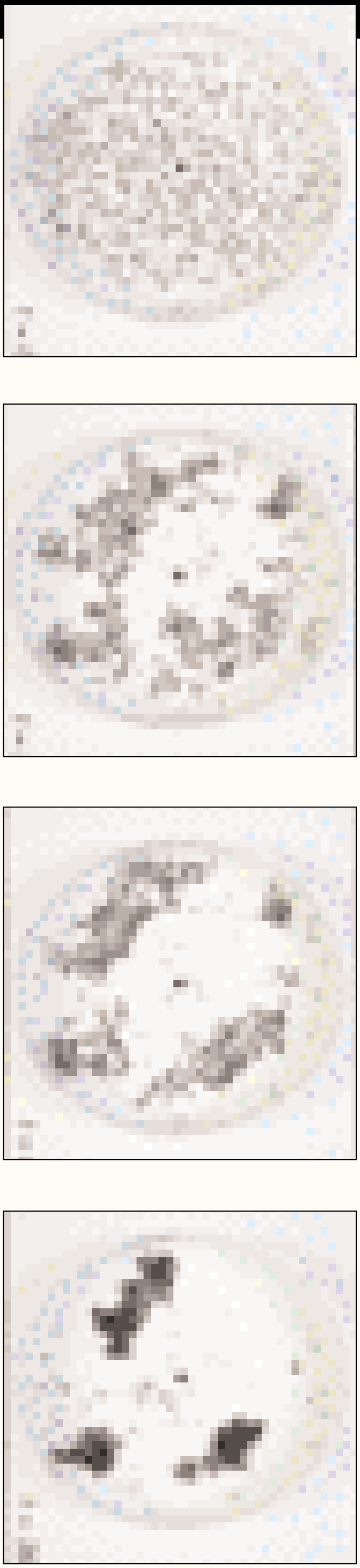
By studying such brood sorting, Erik Lumer of University College London and Baldo Faieta of Interval Research in Palo Alto, Calif., have developed a method for exploring a large database. Imagine that a bank wants to determine which of its customers ismost likely to repay a loan. The problem is that many of the customers have never borrowed money from any financial institution.

But the bank has a large database of customer profiles with attributes such as age, gender, marital status, residential status, banking services used by the customer and so on. If the bank had a way to visualize clusters of people with similar characteristics, loan officers might be able to predict more accurately whether a particular person would repay a loan. If, for example, a mortgage applicant belonged to a group dominated by defaulters, that person might not be a good credit risk.

Because clusters are generally visualized best in two dimensions (higher dimensions make the data difficult for humans to interpret), Lumer and Faieta represent each customer as a point in a plane. So, each client is like a brood item, and software ants can move the clients around, picking them up and depositing them according to the surrounding items. The distance between two customers indicates how similar they are. For the single attribute of age, for instance, shorter distances depict smaller age differences. The artificial ants make their sorting decisions by considering all the different customer characteristics simultaneously. And depending on the bank's objectives, the software could mathematically weigh some of the attributes more heavily than others.

Through this kind of analysis, one cluster might contain people who are about 20 years old, single, mostly living with their parents and whose most popular banking service is interest checking. Another grouping may consist of people who are about 57, female, married (or widowed) and owners of a house with no mortgage.

Of course, banks and insurance companies have typically used similar types of cluster analyses. But the ant-based approach enables the data to be visualized easily, and it boasts one intriguing feature. The number of clusters emerges automatically from the data, whereas conventional methods usually assume a predefined number of groups into which the data are then fit. Thus, antlike



NAMETK *Agencytk*

WORKER ANTS cluster their dead to clean their nest. At the outset of this experiment, 1,500 corpses are located randomly (*top*). After 26 hours, the workers have formed three piles (*bottom*). This behavior and the way in which ants sort their larvae has led to a new type of computer program for analyzing banking data.

another project, a model that was initially introduced to explain how ants cluster their dead and sort their larvae has become the basis of a new approach for analyzing financial data [*see box on page 00*]. And research investigating the flexible way in which honeybees assign tasks could lead to a more efficient method for scheduling jobs in a factory [*see box on page 00*].

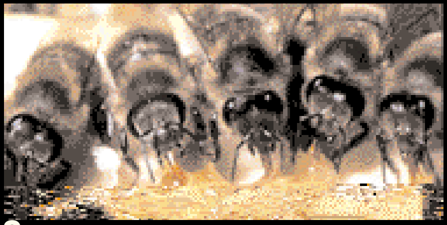
Additional examples abound. Applying knowledge of how wasps construct their nests, Dan Petrovich of the Air Force Institute of Technology in Dayton, Ohio, has designed a swarm of tiny mobile satellites that would assemble themselves into a larger, predefined structure. H. Van Dyke Parunak of the Environmental Research Institute of Michigan in Ann Arbor is deploying a variety of insectlike software agents to solve manufacturing problems—for example, scheduling a complex network of suppliers to a factory. Paul Kantor of Rutgers University has developed a swarm-intelligence approach for finding information over the World Wide Web and in other large networks. Web surfers looking for interesting sites can, if they belong to a "colony" of users, access information in the form of digital pheromones (essentially, ratings) left by fellow members in previous searches.

Indeed, the potential of swarm intelligence is enormous. It offers an alternative way of designing systems that have traditionally required centralized control and extensive preprogramming. It instead boasts autonomy and self-sufficiency, relying on direct or indirect interactions among simple individual agents. Such operations could lead to systems that can adapt quickly to rapidly fluctuating conditions.

But the field is in its infancy. Because researchers lack a detailed understanding of the inner workings of insect swarms, identifying the rules by which individuals in those swarms interact has been a huge challenge, and without such information computer scientists have had trouble developing the appropriate software. In addition, although swarm-intelligence approaches have been effective at performing a number of optimization and control tasks, the systems developed have been inherently reactive and lack the necessary overview to solve problems that require in-depth reasoning techniques. Furthermore, one criticism of the field is that the use of autonomous insectlike agents will lead to unpredictable behavior in the computers they inhabit. This characteristic may actually turn out to be a strength, though, in that it could allow such systems to adapt to solve new, unforeseen problems—a flexibility that traditional software typically lacks.

Many futurists predict that chips will soon be embedded into thousands of mundane objects, from envelopes to trash cans to heads of lettuce. Enabling all these pieces of silicon to communicate with one another in a meaningful way will require novel approaches. As high-technology author Kevin Kelly puts it, "Dumb parts, properly connected into a swarm, yield smart results." The trick, of course, is in the proper connection of all the parts.

Paint Booths with Bee Brains



NAMETK *Agencytk*

HONEYBEES (*top*) perform tasks based on the hive's needs. By studying the way in which these jobs are assigned, scientists hope to develop better ways to program the equipment in an automated factory (*bottom*).

In a honeybee colony, individual insects specialize at certain tasks, depending on their age. Older bees, for example, tend to be the foragers for the hive. But the allocation of tasks is not rigid: when food is scarce, younger nurse bees will forage, too.

Using such a biological system as a model, we have worked with Mike Campos of Northwestern University to devise a technique for scheduling paint booths in a truck factory. In the facility the booths must paint trucks coming out of an assembly line, and each booth is like an artificial bee specializing in one color. The booths can change their colors if needed, but doing so is time-consuming and costly (the equipment must be cleaned, and paint is wasted).

Because scientists have yet to understand exactly how honeybees regulate their division of labor, we made the following assumption: an individual performs the tasks for which it is specialized unless it perceives an important need to perform another function. Thus, a booth with red paint will continue to handle orders of that color unless an urgent job requires a white truck and the other booths, particularly those specializing in white, have much longer queues.

Although this basic rule sounds simplistic, in practice it is very effective. In fact, a honeybeelike system enables the paint booths to determine their own schedules with higher efficiency—specif-

The Authors

ERIC BONABEAU and GUY THÉRAULAZ study the behaviors of social insects and their application in the design of complex systems. Bonabeau is chief scientist at EuroBios in Paris. He received a Ph.D. in theoretical physics and advanced degrees in computer science and applied mathematics from Paris-Sud University. Théraulaz is a research associate at the Laboratoire d'Ethologie et de Psychologie Animale of the Centre National de la Recherche Scientifique (CNRS) in France.

Further Information

Swarm Intelligence: From Natural to Artificial Systems. Eric Bonabeau, Marco Dorigo and Guy Théraulaz. Oxford University Press, 1999.

For more information on ant-based optimization, see iridia.ulb.ac.be/dorigo/ACO/ACO.html on the World Wide Web.

sorting has been effective in discovering interesting commonalities that might otherwise remain hidden.

ically, fewer color changes—than a centralized computer can. And the method is adept at responding to changes in consumer demand. If the number of trucks that need to be painted blue surges unexpectedly, other booths can quickly forgo their specialty colors to accommodate the unassigned vehicles. Furthermore, the system copes easily with glitches. When a paint booth breaks down, other stations compensate swiftly by immediately divvying up the additional load, with idle booths receiving the bulk of the work.