Chapter 6
Swarm Intelligence: The Benefits of Swarms

Dumb parts, properly connected into a swarm, yield smart results.

Kevin Kelly

It is a well-known fact that an individual ant is not very bright and almost blind, but ants in a colony, operating as a team, do remarkable things like effective foraging, optimal brood sorting, or impressive cemetery formation. Many other biological species, such as insects, fish, birds, can present similar intelligent collective behavior although they are composed of simple individuals. At the basis of the increased "intelligence" is the shared "information" discovered individually and communicated to the swarm by different mechanisms of social interaction. In this way, intelligent solutions to problems naturally emerge from the self-organization and communication of these simple individuals. It is really amazing that the seamless coordination of all individual activities does not seem to require *ANY SUPERVISOR*![1]

Swarm intelligence is the emergent collective intelligence of groups of simple individuals, called agents. The individual agents do not know they are solving a problem, but the "invisible hand" of their collective interaction leads to the problem solution. The biological advantages of swarm intelligence for survival of the species in their natural evolution are obvious. Recently, some of the energy savings as a result of the collective behavior of biological swarms have been quantified by proper measurements. For example, a study of great white pelicans has found that birds flying in formation use up to a fifth less energy than those flying solo.[2]

The objective of this chapter is to identify the key benefits of using artificial swarms. The value creation capabilities of swarm intelligence are based on exploring the emerging phenomena driven by social interaction among the individual

[1]Even the famous queen ant has reproductive rather than power-related functions.

[2]H. Weimerskirch, *et al.*, Energy saving in flight formation, *Nature,* 413, (18 October 2001), pp. 697–698, 2001.

A.K. Kordon, *Applying Computational Intelligence*,
DOI 10.1007/978-3-540-69913-2_6,

agents. These emerging phenomena can derive unique routes, schedules, and optimal trajectories, applicable in areas like supply chains, vehicle routing, and process optimization.

Meanwhile, the "dark side" of swarm intelligence is currently a hot topic in science fiction. The famous Michael Crichton novel *Prey* about a swarm of microscopic machines (self-replicating nanoparticles) destroying humans has captured the attention of millions of readers.[3] Unfortunately, the popular negative artistic image of swarm intelligence as a threat to humanity can raise concerns and alienate potential users. One of the objectives of this chapter is to describe the nature of smarm intelligence and to demonstrate the groundlessness of the fears about this emerging technology as the next scientific Frankenstein.

6.1 Swarm Intelligence in a Nutshell

Swarm intelligence is a computational intelligence technique based around the study of collective behavior in decentralized, self-organized systems. The expression "swarm intelligence" was introduced by Beni and Wang in the late 1980s in the context of cellular robotic systems, where many simple robots are self-organized through nearest-neighbor interactions.[4] The research field has grown tremendously since 2000, especially after publishing of the key books, related to the two main development areas, Ant Colony Optimization[5] (ACO) and Particle Swarm Optimization[6] (PSO).

Swarm intelligence systems are typically made up of a population of simple agents interacting locally with one another and with their environment. This interaction often leads to the emergence of global behavior, which is not coded in the actions of the simple agents. Analyzing the mechanisms of collective intelligence that drives the appearance of new complexity out of interactive simplicity requires knowledge of several research areas like biology, physics, computer science, and mathematics. From the point of view of understanding the big implementation potential of swarm intelligence, we recommend the following key topics, shown in the mind-map in Fig. 6.1.

---

[3] M Crichton, *Prey*, HarperCollins, 2002.

[4] G. Beni and J. Wang, Swarm Intelligence, In *Proceedings 7th Annual Meeting of the Robotic Society of Japan*, pp. 425–428, RSJ Press, Tokyo, 1989.

[5] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural Evolution to Artificial System*s, Oxford University Press, 1999.

[6] J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, 2001.

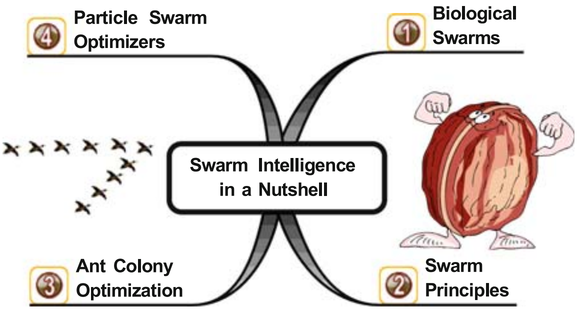


Fig. 6.1 Key topics related to swarm intelligence

6.1.1 Biological Swarms

Swarming as a type of collective interaction is a popular behavior in many biological species. The list includes, but is not limited to: ants, bees, termites, wasps, fish, sheep, and birds. Of special interest are the enormous swarming capabilities of insects. It is a well-known fact that 2% of insects are social. Some swarming insects, like bees, directly create value (or to be more specific, honey) and have been used by humans since ancient time.

The high productivity of bees is based on a unique blend between colony cooperation and effective specialization by division of labor. As a result, food sources are exploited according to quality and distance from the hive, not to mention that the regulation of hive temperature can compete with the most sophisticated digital controllers . . .

The other insects similar to bees – wasps – demonstrate amazing capabilities for "intelligent design" of complex nests. The structure consists of horizontal columns, a protective covering, and a central entrance hole and is built by a flying escadrille of pulp foragers, water foragers and builders.

The building champions among insect swarms, however, are the termites. The building process has two major phases:

- Random walk (uncoordinated phase);
- Coordinated phase.

As a result of this effective self-organization, unique structures, like the termite "cathedral" mound, shown in Fig. 6.2, are built with tremendous speed. The interior "design" is also spectacular with cone-shaped outer walls and ventilation ducts, brood chambers in the central hive, spiral cooling vents, and support pillars.

The famous social insects are ants, which represent about 50% of these biological species. We'll focus on their behavior in more detail in Sect. 6.1.3 but here are some highlights. First, we share a little known fact that the total weight of all ants added together is equal to the total weight of humans (the average weight of an ant is between 1 and 5 mg). However, ants began their evolutionary battle for survival 100 million years ago, much earlier than our ancestors.

Fig. 6.2 A termite "cathedral" mound produced by a termite colony[7]

The ants' efficiency through social interaction continues to surprise researchers.[8] Examples of such are: capabilities like organizing "highways" to and from their foraging sites by leaving pheromone[9] trails, forming chains from their own bodies to create "bridges" to pull and hold leaves together with silk, and the almost perfect division of labor between major and minor ants. Some ant colonies have networks of nests several hundreds of meters in span. The most advanced army allocation pattern, shown in Fig. 6.3, belongs to the tropical ant *Eciton burchelli*. It includes as many as 200,000 blind workers and its structure consists of a 15 m-wide swarm front, a dense ant phalange 1 meter behind, and a complex trail that converges to a single straight line "highway" to the bivouac. The Art of War of this army of ants doesn't need generals.

Fish schooling is another known form of swarm intelligence (see Fig. 6.4). Schools are composed of many fish of the same species moving in more or less harmonious patterns throughout the water. A very prevalent behavior, schooling is exhibited by almost 80% of the more than 20,000 known fish species during some phase of their life cycle.

Why do fish school? One of the key reasons is that some species of fish secrete a slime that helps to reduce the friction of water over their bodies. In addition, the fish swim in precise, staggered patterns when traveling in schools and the motion

---

[7]http://www.scholarpedia.org/article/Swarm_intelligence

[8]A very interesting book on this topic is D. Gordon, *Ants at Work: How an Insect Society is Organized*, W. Norton, NY, 1999.

[9]A pheromone is a chemical used by animals to communicate.

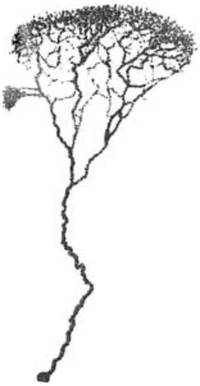


Fig. 6.3 Forging patterns of an army of Eciton burchelli[10]

Fig. 6.4 Fish school

of their tails produces tiny currents called vortices.[11] Each individual can use the tiny whirlpool of its neighbor to assist in reducing the water's friction on its own body.

[10]http://www.projects.ex.ac.uk/bugclub/raiders.html

[11]Swirling motions similar to little whirlpools.

Another reason for schooling is the increased safety against predators. A potential predator looking for a meal might become confused by the closely spaced school, which can give the impression of one vast fish.

We'll finish this section by answering the generic question: Why do animals swarm? The four key reasons, according to biologists, are: (1) defense against predators by enhanced predator detection and minimizing the chances of being captured; (2) improved foraging success rate; (3) better chances to find a mate; and (4) decrease of energy consumption.

6.1.2 Principles of Biological Swarms

The next topic of interest is defining the generic principles behind the different forms of swarm behavior. It will lead to the design of artificial swarms. Firstly, let's define the key characteristics of a swarm as:

- Distributed: no central data source;
- No explicit model of the environment;
- Perception of the environment (sensing capability);
- Ability to change the environment.

Secondly, we'll focus on the key issue of self-organization of biological swarms. It is the complexity and sophistication of self-organization that allows functioning with no clear leader. The essence of self-organization is the appearance of a structure without explicit external pressure or involvement.

The obvious result from self-organization is the creation of different structures, such as social organization based on division of labor, foraging trails, and all of these remarkable designs of nests.

Thirdly, the mechanism of the unique indirect way of communication of biological swarms, called stigmergy, will be discussed. Stigmergy is defined as the indirect interaction of two individuals when one of them modifies the environment and the other responds to the new environment at a later time. Since no direct communication takes place between individuals, information is communicated through the state or changes in the local environment. In some sense, environmental modification serves as external memory and the work can be continued by any other individual. Stigmergy is the basis of coordination by indirect interaction, which in many cases for biological swarms is more appealing than direct communication.

The final topic in this section is the basic principles of swarm intelligence, as defined by Mark Millonas from Santa Fe Institute:[12]

---

[12]M. Millonas. Swarms, phase transitions, and collective intelligence. In C.G. Langton (Ed.), *Artificial Life III*, pp. 417-445, Santa Fe Institute Studies in the Sciences of the Complexity, Vol. XVII, Addison-Wesley, 1994.

- *Proximity Principle*: individuals should be able to interact so as to form social links.
- *Quality Principle*: individuals should be able to evaluate their interactions with the environment and one another.
- *Diverse Response Principle*: the population should not commit its activities along excessively narrow channels.
- *Stability Principle*: the population should not change its mode of behavior every time the environment changes.
- *Adaptability Principle*: the population must be able to change behavior mode when necessary.

From that perspective, a swarm system is composed of a set of individuals which interact with one another and the environment. Swarm intelligence is defined as an emerging property of the swarm system as a result of its principles of proximity, quality, diversity, stability, and adaptability.[13]

There are two key directions in research and applied swarm intelligence: (1) Ant Colony Optimization (ACO), based on insect swarm intelligence; and (2) Particle Swarm Optimizers (PSO), based on social interaction in bird flocking. Both approaches will be discussed in the next two sections.

6.1.3 Ant Colony Optimization

Individual ants are simple insects with limited memory and capable of performing simple actions. However, an ant colony generates a complex collective behavior providing intelligent solutions to problems such as: carrying large items, forming bridges, finding the shortest routes from the nest to a food source, prioritizing food sources based on their distance and ease of access, sorting corpses. Moreover, in a colony each ant has its prescribed task, but the ants can switch tasks if the collective needs it. For example, if part of the nest is damaged, more ants do nest maintenance work to repair it

One of the fundamental questions is: How do ants know which task to perform? When ants meet, they touch with their antennas. It is a well-known fact that these are organs of chemical perception and the ant can perceive the colony-specific odor from all members of the nest. In addition to this odor, ants have an odor specific to their task, because of the temperature and humidity conditions in which it works, so that an ant can evaluate its rate of encounter with ants of a certain task. In addition, the pattern of ant influences the probability of performing a specific task.

How can ants manage to find the shortest path? The answer from biology is simple – by applying the stigmetry mechanism of indirect communication based on pheromone deposition over the path they follow. The scenario is as follows:

---

[13] L. de Castro, *Fundamentals of Natural Computing*, Chapman & Hall, 2006.

– An isolated ant moves at random, but when it finds a pheromone trail, there is a high probability that this ant will decide to follow the trail.

– An ant foraging for food deposits pheromone over its route. When it finds a food source, it returns to the nest reinforcing its trail.

– Other ants have greater probability to start following this trail and laying more pheromone on it.

– This process works as a positive feedback loop system because the higher the intensity of the pheromone over a trail, the higher the probability of an ant to start traveling through it.

The short-path ant algorithm is demonstrated in Fig. 6.5 in the case of two competing routes, one of which is significantly shorter. Let's assume that in the initial search phase, an equal number of ants moves to both routes. However, the ants on the short path will complete the travel more times and thereby lay more pheromone over it. The pheromone concentration on the short trail will increase at a higher rate than on the long trail, and in the advanced search phase the ants on the long route will choose to follow the short route (the amount of pheromone is proportional to the thickness of the routes in Fig. 6.5). Since most ants will no longer travel on the long route, and since the pheromone is volatile, the long trail will start evaporating. In the final search phase only the shortest route will remain.

Surprisingly, this simple algorithm is at the basis of a method for finding optimal solutions in real problems, called Ant Colony Optimization (ACO). Each artificial

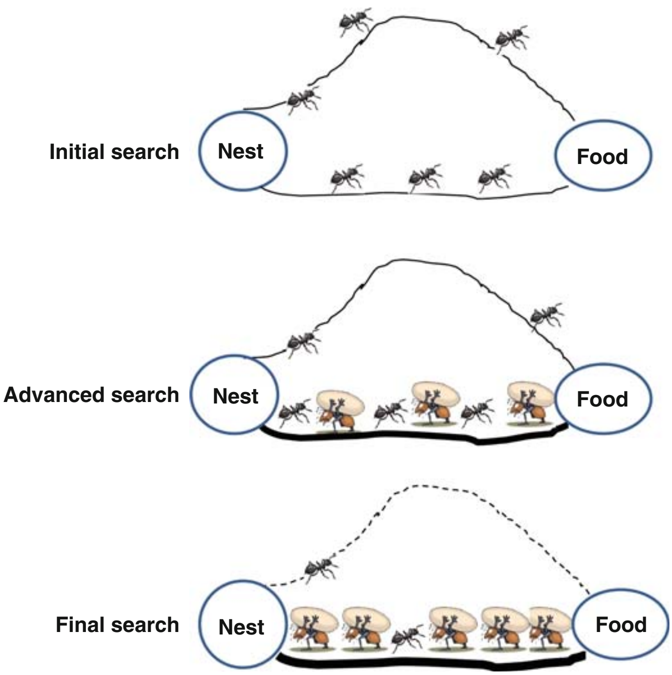


Fig. 6.5 Ant route handling

ant is a probabilistic mechanism that constructs a solution to the problem, using artificial pheromone deposition, heuristic information about pheromone trails and a memory for already visited places.

In Ant Colony Optimization, a colony of artificial ants gradually constructs solutions for a defined problem, using artificial pheromone trails, which are modified accordingly during the algorithm. In the solution construction phase, each ant builds a problem-specific solution – for example, selection of the next route for the supply chain.

The choice of solution fragment by an artificial ant at each step of the construction stage is proportional to the amount of artificial pheromone deposited on each of the possible solutions. In the next step of the algorithms, after all ants have found a solution, the pheromone deposits on each solution fragment are updated. The high-quality solution fragments are supported by stronger pheromone reinforcement. After several iterations, better solution fragments are more frequently used by the artificial ant colony and the opposite, less successful, solutions gradually disappear. The pheromone trails also evaporate during the update in order to forget the least used solutions.

6.1.4 Particle Swarm Optimizers

In contrast to the insect-driven ant colony optimization, the second key direction in swarm intelligence, invented by Jim Kennedy and Russ Eberhart in the mid-1990s,[14] is inspired mostly by the social behavior of bird flocking and fish schooling.

One of the key questions in analyzing flock behavior is: How does a large number of birds produce a seamless, graceful flocking dance, while often, but suddenly changing direction, scattering and regrouping? The impression is that even though the individual birds change the shape and the direction of their motion, they appear to move as a single coherent organism. The analyses of flock behavior of various types of birds have led to defining the main flock principles, as shown in the mind-map on Fig. 6.6 and summarized below.

Flock principles:[15]

1. *Velocity Matching*: attempt to match velocity with nearby flock mates.
2. *Flock Centering*: attempt to stay close to nearby flock mates.
3. *Collision Avoidance*: avoid colliding with nearby flock mates.
4. *Locality*: its nearest flock mates only influence the motion of each bird, i.e., vision is the most important sense for flock organization.

---

[14] The original paper is: J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of the IEEE Int. Conf. on Neural Networks*, Perth, Australia, pp. 1942–1948, 1995.

[15] S. Das, A. Abraham, and A. Konar, Swarm intelligence algorithms in bioinformatics, In *Computational Intelligence in Bioinformatics*, A. Kelemen, *et al.* (Eds), Springer, 2007.

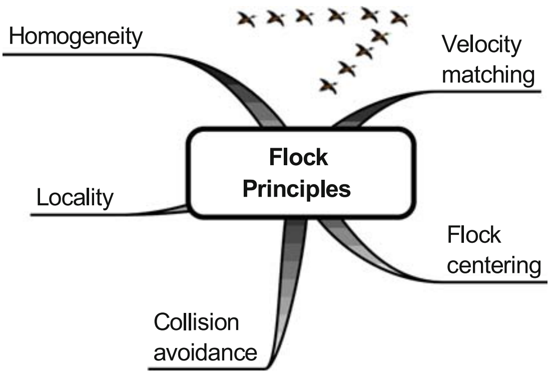


Fig. 6.6 Key flock principles

5. *Homogeneity*: each bird in the flock has the same behavior. The flock moves without a leader, even in cases when temporary leaders appear.

The defined flock principles are at the core of the Particle Swarm Optimization (PSO) algorithm. The analogy is very direct: in PSO, each single solution is like a 'bird' in the search space, which is called a "particle". A selected number of solutions (particles) form a flock (swarm) which flies in a D-dimensional search space trying to uncover better solutions. For the user the situation recalls the simulation of bird flocking in a two-dimensional plane. Each particle is represented by its position on the XY plane as well as by its velocity (V_x as the velocity component on the X-axis and V_y as the velocity component on the Y axis). All particles in the swarm have fitness values which are evaluated by a defined fitness function to be optimized, and have velocities which direct the flying of the particles. (The particles fly through the problem space by following the particles with the best solutions so far.)

Each particle also has a memory of the best location in the search space that it has found (*pbest*) and knows through social interaction the best location found to date by all the particles in the flock (*gbest* or *lbest*). The way the best location found is obtained depends on the swarm topology. There are different neighborhood topologies used to identify which particles from the swarm can influence the individuals. The most common ones are known as the *gbest* or fully connected topology and *lbest* or ring topology and are illustrated in Fig. 6.7.

In the *gbest* swarm topology, shown in Fig. 6.7a, the trajectory of each particle is influenced by the best individual found in the entire swarm (shown as a big bird). It is assumed that *gbest* swarms converge fast, as all the particles are attracted simultaneously to the best part of the search space. However, if the global optimum is not close to the best particle, it may be impossible for the swarm to explore other areas and, consequently, the swarm can be trapped in local optima.

In the *lbest* swarm topology, shown in Fig. 6.7b, each individual is influenced by a smaller number of its neighbors (which are seen as adjacent members of the swarm ring). Typically, *lbest* neighborhoods comprise two neighbors: one on the

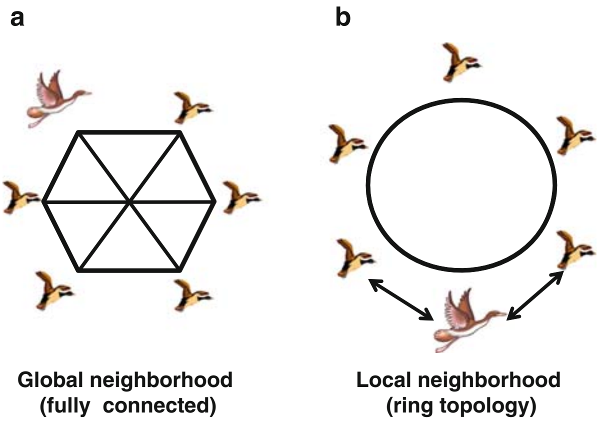


Fig. 6.7 Graphical representation of *gbest* (*a*) and *lbest* (*b*) swarm topologies

right side and one on the left side (a ring lattice). This type of swarm will converge slower but can locate the global optimum with a greater chance.

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating each generation. At each generation, each particle is updated by the following two "best" values – of the particle itself and of the neighborhood. The first one is the best previous location (the position giving the best fitness value) a particle has achieved so far. This value is called *pbest*. At each iteration the *P* vector of the particle with the best fitness in the neighborhood, designated *lbest* or *gbest*, and the *P* vector of the current particle, are combined to adjust the velocity along each dimension, and that velocity is then used to compute a new position for the particle. The two equations, for particle velocity and position update, that drive PSO are given below:

new velocity

$$v_{k+1}^{i} = w\underbrace{v_k^{i}}_{\text{current motion}} + c_1\,rand\,\underbrace{\frac{(p^{i} - x_k^{i})}{\Delta t}}_{\text{particle memory influence}} + c_2\,rand\,\underbrace{\frac{(p_k^{g} - x_k^{i})}{\Delta t}}_{\text{swarm influence}}$$

inertia factor

0.4 to 1.4

self-confidence

1.5 to 2

self-confidence

2 to 2.5

New position

New velocity

$$x_{k+1}^{i} = x_k^{i} + v_{k+1}^{i}\ .$$

Current position

New velocity (which denotes the amount of change) of the *i*-th particle is determined by three components:

(1) momentum or current motion – the current velocity term to push the particle in the direction it has traveled so far;
(2) cognitive component or particle memory influence – the tendency to return to the best position visited so far by the *i*-th particle;
(3) social component or swarm influence – the tendency to be attracted towards the best position found in its neighborhood either by the ring topology *lbest* or by the star topology *gbest*.

A visual interpretation of the PSO algorithm is given in Fig. 6.8.

Let's assume that a particle (visualized by a bird) has a current position $X(k)$ at time k and its current velocity is represented by the vector $V(k)$. The next position of the bird at time $k+1$ is determined by its current position $X(\mathrm{k})$ and the next velocity $V(k+1)$. The next velocity, according to the new velocity PSO equation, is a vector blending of the current velocity $V(k)$ with the acceleration component towards the swarm-best (represented by the big bird) with its velocity of V_{gbest} and the other acceleration component towards the particle best with its velocity of V_{pbest}. As a result of these velocity adjustments, the next position $X(k+1)$ is closer to the global optimum.

The other parameters in the PSO algorithm have the following interpretation. The inertia or momentum factor w controls the impact of the particle's previous velocity on its current velocity and plays a balancing role between encouraging a more intensive local search of already discovered perspective regions (low w values) and exploring new diverse areas (high w values).

The purpose of the two random numbers *rand* in the PSO new velocity equation is to ensure that the algorithm is stochastic and neither the cognitive nor the social components are dominant. The direct control of the influence of the social and

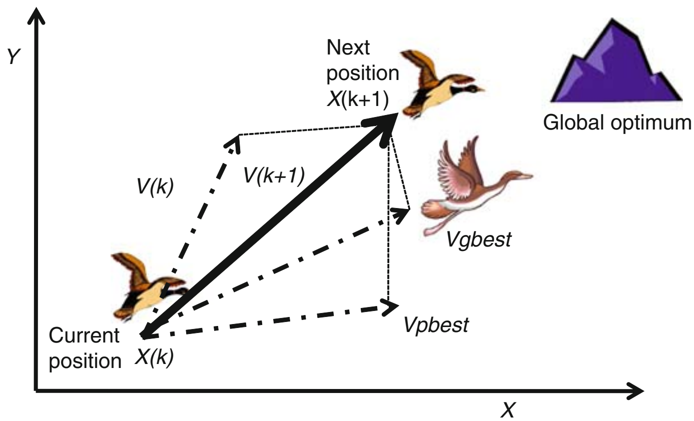


Fig. 6.8 Visualization of particle position update diagram

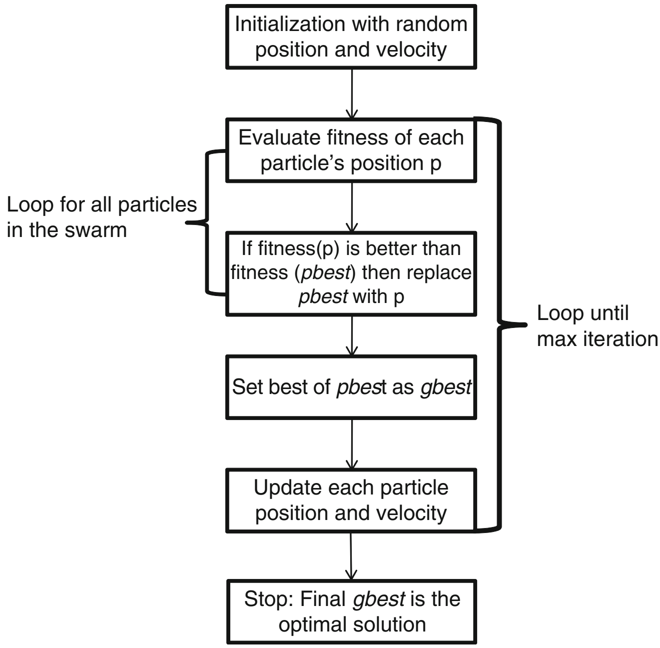


Fig. 6.9 A flowchart of the PSO algorithm

cognitive components is done by the c_1 and c_2 weight coefficients (called also self-confidence). Low values of these coefficients allow each particle to explore far away from already uncovered high-fitness points, high values of these parameters push towards more intensive search of high fitness regions.

Another important feature of the PSO algorithm is that particle velocities are clamped to the range [-V_{max},V_{max}] which serves as a constraint to control the global exploration ability of the particle swarm. Thus, the likelihood of particles leaving the search space is reduced. Note that this is not to restrict the values of Xi within the range [-V_{max}, V_{max}]; it only limits the maximum distance that a particle will move during one iteration.

The flow chart of the PSO algorithm, which is self-explanatory, is given in Fig. 6.9.

6.2 Benefits of Swarm Intelligence

The value creation capabilities of swarm intelligence based on social interactions may generate tremendous benefits, especially in the area of nontrivial optimization of complex problems. The specific advantages of using swarm intelligence by both ACO and PSO methods are discussed in this section. Firstly, we'll begin by clarifying some similarities and differences between swarm intelligence and evolutionary computation.

6.2.1 Comparison Between Evolutionary Computation and Swarm Intelligence

Swarm intelligence, in general, and PSO in particular, shares many common features with evolutionary algorithms, especially with genetic algorithms (GA). For example, both algorithms (GA and PSO) start with a group of a randomly generated population. Both have fitness values to evaluate the population. Both update the population and search for the optimum with random techniques. Both systems do not guarantee finding the global optimum.

However, PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is an important feature of the algorithm. Compared with genetic algorithms, the information sharing mechanism in PSO is significantly different. In GAs, chromosomes share information with each other. So the whole population moves like a one group towards an optimal area. In PSO, only the swarm leader (*gbest* or *lbest*) spreads the information to others. It is a one-way information sharing mechanism. Compared with GA, in most cases all the particles tend to converge to the best solution quickly even in the local version.

Other differences between swarm intelligence (PSO) and evolutionary algorithms can be defined as:

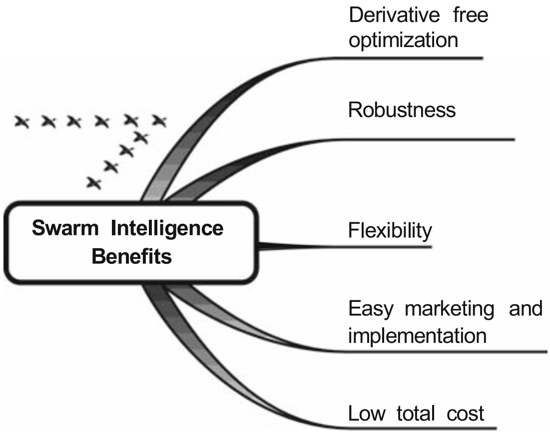
- A particle has a position (contents of a candidate solution) and a velocity whilst an individual in evolutionary algorithms typically has just the contents of a candidate solution.
- In evolutionary algorithms individuals compete for survival and most "die" while the population of the swarm is constant.
- The key difference is based on the driving forces of novelty generation. In the case of evolutionary algorithms the new solutions emerge from the strong struggle for high fitness. In the case of swarm intelligence, the novelty is generated by social interaction between the individuals.

6.2.2 Benefits of Swarm Intelligence Optimization

The key benefits of applying both ACO and PSO are captured in the mind-map, shown in Fig. 6.10 and discussed below.

- *Derivative-Free Optimization* – The search for an optimal solution in swarm intelligence is not based on functional derivatives but on different mechanisms

Fig. 6.10 Key benefits from swarm intelligence



of social interaction between artificial individuals. In this way the chances of being entrapped in local minima are significantly reduced (but not eliminated!).

- *Robustness* – The population-based ACO and PSO algorithms are more protective towards individual failure. The poor performance of even several members of the swarm is not a danger for the overall performance. The collective behavior compensates the laggards and the optimum solution is found independently of the variations in individual performance.
- *Flexibility* – Probably the biggest benefit from swarm intelligence is its capability to operate in a dynamic environment. The swarm can continuously track even for fast-changing optima. In principle, there is no significant difference in functioning of the algorithm in steady-state or in dynamic mode. In the case of classical methods, different algorithms and models are required for these two modes.
- *Easy Marketing and Implementation* – The principles of biology-inspired swarm intelligence are easy to communicate to a broad audience of potential users and there is no need for a heavy math or statistical background. In addition, the implementation of both ACO and especially PSO on any software environment is trivial. The tuning parameters are few and easy to understand and adjust. In some cases, implementing PSO can even be transparent for the final user and be a part of the optimization options of a larger project.
- *Low Total Cost* – In summary, the low marketing and implementation cost, as well as potentially low maintenance cost due to the built-in adaptability in changing operating conditions, result in low total-cost-of-ownership.

6.3 Swarm Intelligence Issues

PSO have two major algorithmic drawbacks. The first drawback is that PSO usually suffers from premature convergence when problems with multiple optima are being optimized. The original PSO is not a local optimizer and there is no guarantee that the solution found is a local optimum. At the basis of this problem is that, for the *gbest* PSO, particles converge to a single point, which is on the line between the global best and the personal best positions.

The second drawback of PSO is that its performance is very sensitive to parameter settings. For example, increasing the value of the inertia weight, w, will increase the speed of the particles resulting in more exploration (global search) and less exploitation (local search) and vice versa. Tuning the proper inertia is not an easy task and is problem-dependent.

Beyond these specific technical issues, both ACO and PSO lack a solid mathematical foundation for analysis, especially for realistic algorithm convergence conditions and a generic methodology for parameter tuning. An addition, there are some questions about the "dark side" of swarm intelligence. On the technical front, questions are asked about some serious issues, such as the nature of predictability in distributed bottom-up approaches, the efficiency of the emergent behavior, and the dissipative nature of self-organization. On the social front, there is a growing resistance towards two potential application areas of swarm intelligence – military/law enforcement and medical. In the efforts of fighting the war on terror, the first application area has recently been explored very actively. As a result, designing and using flocks of flying and interacting smart micro-robots with miniature cameras for surveillance and in military actions is not science fiction anymore. Very soon it may change the nature of war and intelligence. However, the perspective of a new technological Big Brother as continuously tracking and spying smart swarms is chilling.

Even scarier looks the other big potential application area – using miniature nanoswarms for fighting diseases, especially cancer. The initial inspiring idea of designing a swarm of nanoparticles carrying specific medicine and moving it to a target area with cancer cells was diverted in the negative direction by the novel *Prey*. The potential for internal destruction of the human body by a nanoswarm killer, so vividly described in fiction, creates an attitude to prevent this happening in nonfiction.

6.4 How to Apply Swarm Intelligence

Due to the simple algorithms, implementing swarm intelligence on any software environment is trivial. However, applying ACO and PSO require different tuning parameters and problem formulation and will be discussed separately.

6.4.1 When Do We Need Swarm Intelligence?

The key capabilities of swarm intelligence that may create value are shown in the mind-map in Fig. 6.11. No doubt the most valuable feature of swarm intelligence is its potential in optimization. However, both swarm intelligence methods have

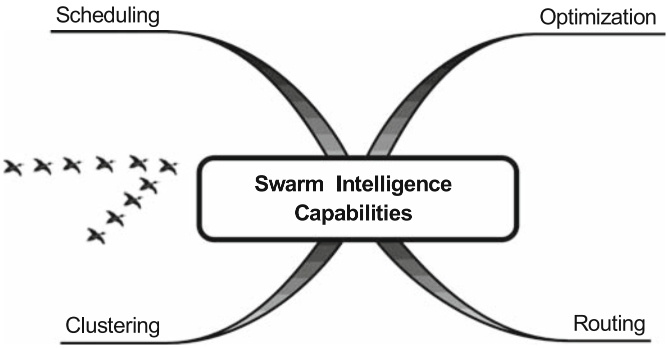


Fig. 6.11 Key capabilities of swarm intelligence

different optimization markets. ACO algorithms are in general more suitable to combinatorial optimization. The goal of combinatorial optimization is to find values for discrete variables (structures) that optimize the value of an objective function. Let's not forget that at the basis of ACO are artificial ants walking on an artificial graph. Graphs are a typical example of discrete structures.

On the other hand, PSO algorithms are in general more suitable to functional optimization where the goal is to find the optimal value of a certain function of real-valued variables. PSO shines especially in solving optimization problems which are unsuitable or infeasible for analytical or exact approaches. Examples are nasty functions with multiple optima, hard to model nonstationary environments, distributed systems with limited measurements, and problems with many variables and sources of uncertainty. Of special importance is the capability of PSO for dynamic optimization, an area where it has a competitive advantage versus the other methods, especially GA.

Many real-world applications of swarm intelligence are driven by the unique capability of ACO to use artificial ants for optimal routing. Most of the successful routing applications are based on different versions of the ant-foraging models and are specific for each implementation area, such as routing in telephone networks, data communication networks, and vehicles. However, beyond some level of complexity, there is a limitation since routing algorithms are generally difficult to analyze either mathematically or visually. Unfortunately, convergence to the optimal solution is not guaranteed, the speed of adaptation to fast changes could be unacceptable, and oscillatory behavior of the algorithm cannot be excluded.

Both ACO and PSO are capable of performing sophisticated clustering. An example is the PSO clustering, which overcomes some of the limitations of the popular clustering algorithms like K-means.[16] For example, K-means has no "global view" of the clustering solution: each centroid (cluster center) moves to the centre of its assigned examples, regardless of other centroids. In addition,

[16]K-means is an algorithm for data partitioning into *K* clusters so that the within-cluster distance is minimized.

different initial centroids (generated at random) can lead to different clustering results; so it is recommended to run K-means many times with a different set of initial conditions at each run and to check if (almost) all runs lead to similar results.

Fortunately, PSO handles these limitations and improves clustering efficiency significantly. PSO moves the centroids[17] according to its global search procedure, i.e. each particle has a "global view" of its entire clustering solution and the fitness function takes into account the positions of all centroids in the particle. The PSO population also contains a number of different sets of centroids. This is similar to multiple runs of the K-means algorithm but, instead of multiple independent runs, during the PSO search the particles "communicate" with each other, allowing them to share information about areas of the search space with high fitness.

Using artificial ants for scheduling and task allocation is another competitive capability of swarm intelligence. Examples are scheduling paint booths in a truck factory and optimal ordering of pickers at a large distribution center of a major retail chain.[18]

6.4.2 Applying Ant Colony Optimization

The generic application sequence for ACO is shown in Fig. 6.12. It begins with one of the most time-consuming steps of defining an appropriate representation of the problem. It includes specifying the components that an ant will use to incrementally construct a candidate solution and especially paying attention to enforcing the construction of valid solutions. For example, in the case of finding optimal routes, candidate solutions could be the distances to specific locations.

In the same way as data collection is critical for applying fuzzy, machine learning, and evolutionary computation systems, selecting a representative test case is decisive for ACO implementation. On the one hand, the selected case must drive the optimization algorithm development and adjustment. On the other hand, it has to cover the broadest possible conditions to validate the ACO performance.

The next application step is defining a problem-dependent heuristic function (η) that measures the quality of each component that can be added to a partial candidate solution. It has to specify how to update the amount of pheromone (τ) associated with each component in a path followed by an ant. Usually pheromone increases in proportion to the quality of the path (solution). As a result, a probabilistic transition rule based on the value of the heuristic function η and the current amount of pheromone τ associated with each candidate solution component is defined.

---

[17] A particle is defined as a set of centroids.

[18] E. Bonabeau and C. Meyer, Swarm intelligence: A whole new way to think about business, *Harvard Business Review*, May 2001.

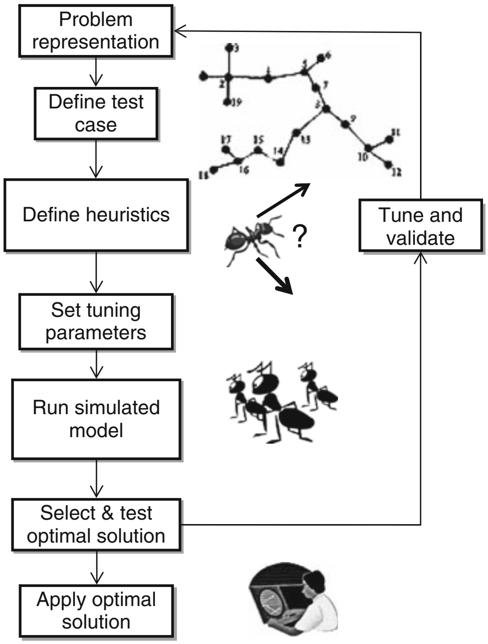


Fig. 6.12 Key steps in applying ant colony optimization

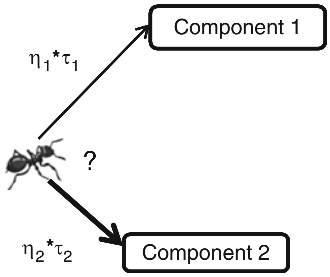


Fig. 6.13 Each ant selects the next component based on the product of the current amount of pheromone τ and a problem-specific heuristic function η

This rule is used to decide which solution component is chosen to be added to the current partial solution. Typically, the probability of choosing a component *i* is proportional to the product $\eta_i \times \tau_i$. An illustration of such a rule is shown in Fig. 6.13.

The next step in the ACO application sequence is setting the tuning parameters. Some of these parameters, such as the number of ants, the stopping criteria, based either on accuracy or prescribed number of iterations, or the number of repetitive runs, are generic. The rest of the tuning parameters, such as pheromone ranges and rate of evaporation are algorithm-specific.

Usually selection of an ACO algorithm requires several simulation runs on the test case with refined adjustment of the tuning parameters until an acceptable performance is achieved. Then the developed algorithm can be used on other similar applications.

6.4.3 Applying the Particle Swarm Optimizer

The PSO application sequence is shown in Fig. 6.14. It begins with the definition of a PSO particle. In the same way as defining the GA chromosome within the context of the solved problem is critical for application success, implementing PSO successfully depends on mapping the problem solution into the PSO particle. For example, if the objective is to optimize some process variable, such as to maximize ethylene production in a cracking furnace, which depends on key variables like steam-to-naphtha ratio, the outlet furnace temperature, and the outlet furnace pressure, the PSO particle **p** is defined as a vector with three components (the three related process variables) and a value of the produced ethylene. It is very important to have estimates of the ranges of each component.

The second step of the PSO application sequence includes the preparation of a representative test case that will help to develop, tune, and validate the derived solution. Ideally, it is preferable to use data and knowledge for the full range of expected operation of the solution.

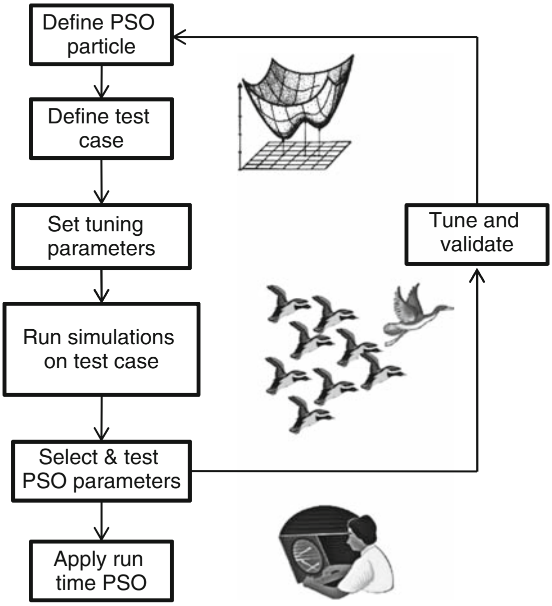


Fig. 6.14 Key steps in applying particle swarm optimization

A typical PSO setting includes the following parameters:

- Neighborhood structure: the global version is faster but might converge to a local optimum for some problems. The local version is a little bit slower but with lower probability to be trapped into a local optimum. One can use the global version to get a quick result and use the local version to refine the search.
- Number of particles in the population: the typical range is from 20 up to 50. Actually for most of the problems 10 particles are sufficient to get good results. For some difficult or special problems, one can try 100 or 200 particles as well.
- Dimension of particles: this is determined by the problem to be optimized.
- Range of particles: this is also determined by the problem to be optimized; you can specify different ranges for different dimensions of particles.
- $V_{\max}$: this determines the maximum change one particle can take during one iteration. Usually the range of $V_{\max}$ is related to the upper limit of $X_{\max}$.
- Learning factors: c_1 and c_2 are usually equal to 2. However, other settings were also used in different references. But usually c_1 equals c_2 and is in the range $[0, 4]$.
- Inertia weight w: usually is in the range 0.4 to 1.4.
- Maximum number of iterations: problem-dependent, the typical range is 200 up to 2000 iterations.

Due to the stochastic nature of PSO, it is recommended that the algorithm is run for at least 20 simulations for a given set of parameters. The final solution is selected after an exhaustive tuning and validation in all possible conditions. It could be applied as a run-time optimizer for similar problems.

6.4.4 Applying the Particle Swarm Optimizer: An Example

The PSO application sequence is illustrated with a simple example of optimizing a function with multiple optima:

$$y = \sin(n * x_1) + \cos(n * x_1 * x_2) + n * (x_1 + x_2)$$

where the number of optima n can be a setting parameter.

The particle is defined as the maximum value of y with dimensionality of two $[x_1, x_2]$. The ranges of the both dimensions are between -1 and $+1$. The PSO tuning parameters are as follows: population size = 50, $V_{max} = +1$, $c_1 = 3$, $c_2 = 1$, $w = 0.73$, maximum number of iterations = 250.

The objective of the study is to explore the PSO performance when the search space becomes complex due to the high number of multiple optima. Of special interest are the cases if PSO can distinguish between two geographically close optima with small differences relative to the global optimum. In order to validate the reproducibility of the results the PSO was run 20 times.

The PSO reliably identified the global optimum up to the case with 32 multiple optima. A 2D grid of the fitness landscape with 16 optima is shown in Fig. 6.15.

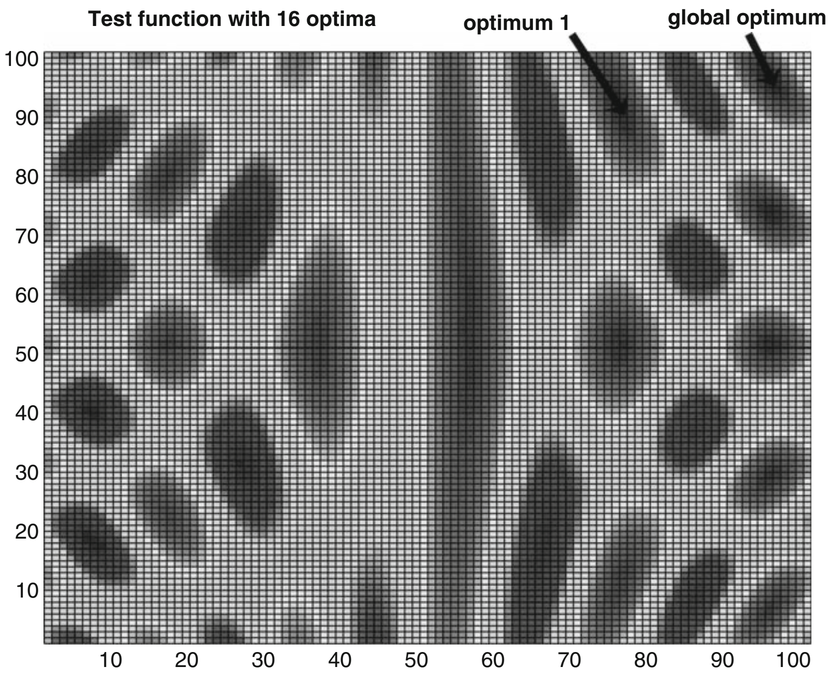


Fig. 6.15 A 2D grid of a function with 16 optima. The differences between optimum 1 and the global optimum (shown with arrows) is very small

Optimum 1 is geographically close to the global optimum and the difference in their fitness is small. A typical distribution of the global solutions *gbest*, generated during the PSO run of a function with 16 optima is shown in Fig. 6.16. The distribution reliably represents the fitness landscape, shown in Fig. 6.15, and identified the global optimum in 100% of the cases.

However, the results for optimization of a function with 32 optima are not so impressive. The 2D grid of the fitness landscape with 32 optima is shown in Fig. 6.17. In this case optimum 1 is geographically closer to the global optimum and the difference in their fitness is almost negligible. In 60% of the runs PSO cannot identify correctly the global optimum and converges to optimum 1 (see Fig. 6.18). Only after tuning the parameters by increasing the population size to 100 and the number of maximal iteration to 500, ca PSO identify the global optimum with 90% success rate (see Fig. 6.19).

6.5 Typical Swarm Intelligence Applications

The application record of swarm intelligence is not as impressive as that of the more-established computational intelligence methods. However, the speed of adoption in real-world applications, especially of PSO, is growing. The key selected

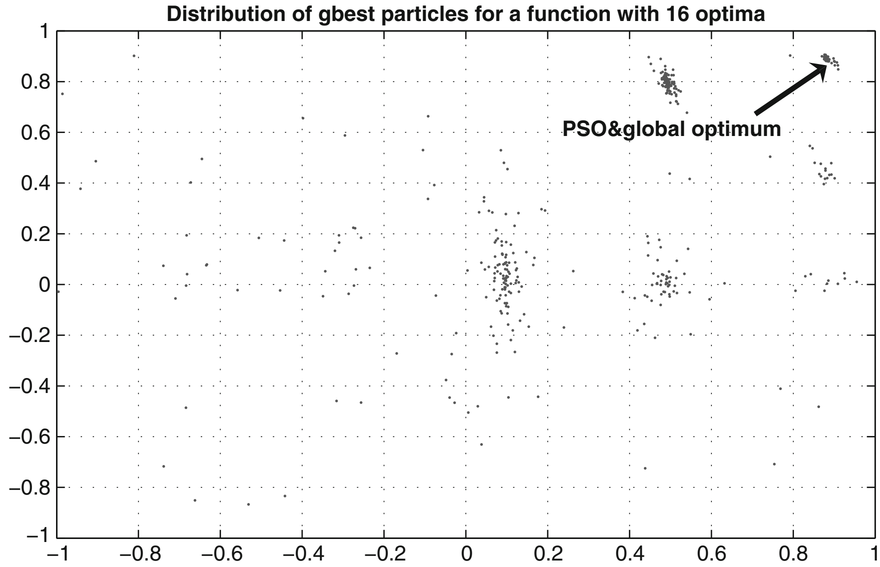


Fig. 6.16 Distribution of *gbest* particles after the final iteration in case of a function with 16 optima

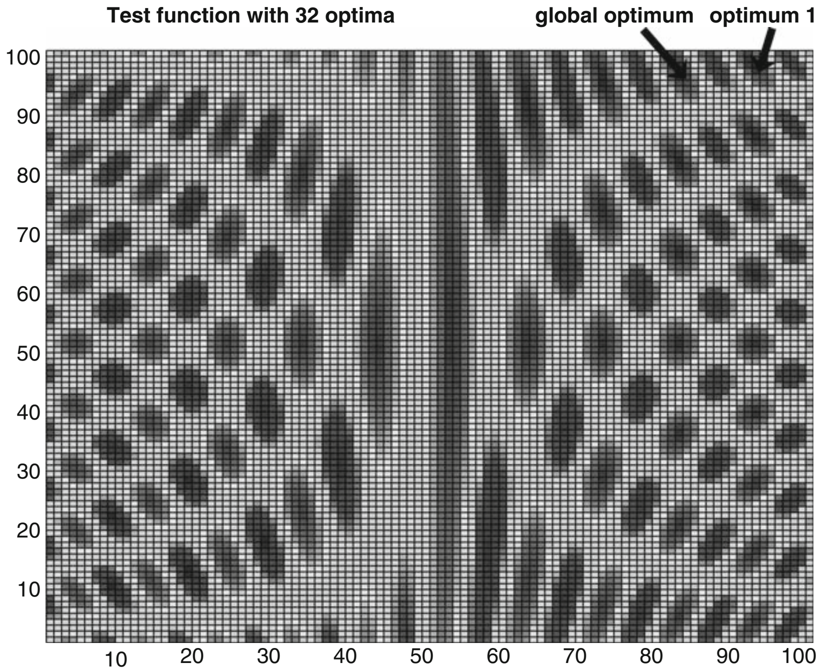


Fig. 6.17 A 2D grid of a function with 32 optima. The differences between optimum 1 and the global optimum (shown with arrows) is negligible

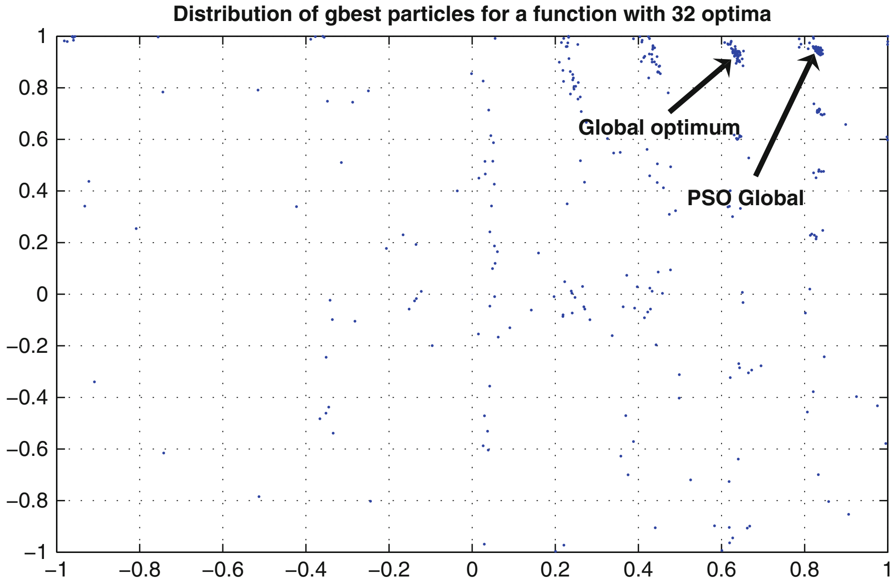


Fig. 6.18 Distribution of *gbest* particles after the final iteration in case of a function with 32 optima and different global optimum and PSO global solution

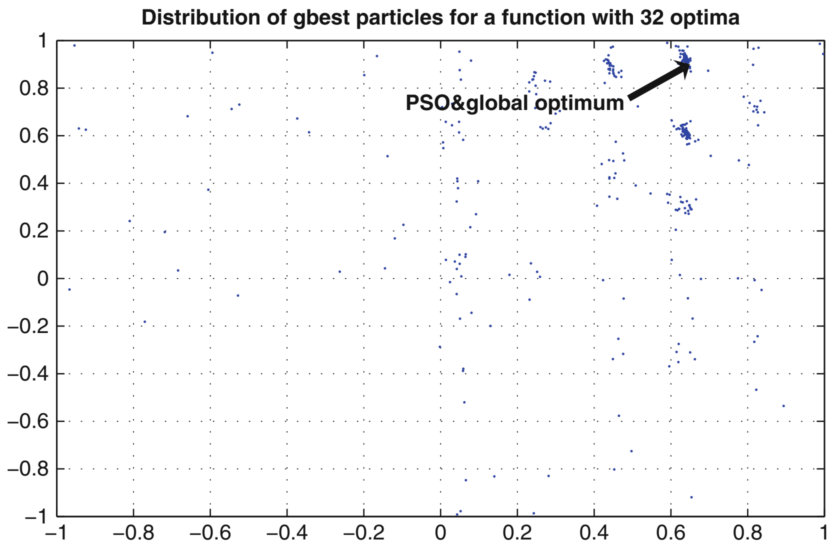


Fig. 6.19 Distribution of *gbest* particles after the final iteration in the case of a function with 32 optima and convergence of the PSO global solution to the global optimum

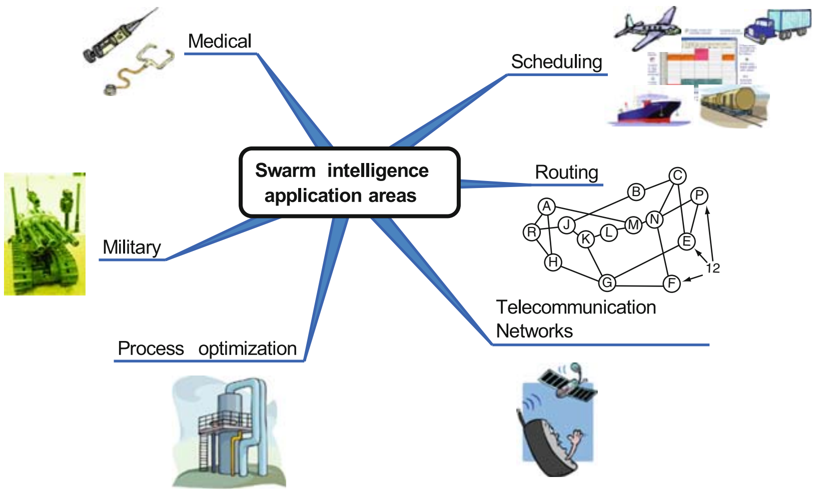


Fig. 6.20 Key swarm intelligence application areas

application areas of swarm intelligence are shown in the mind-map in Fig. 6.20 and discussed below.

- *Scheduling* – The unique features of ACO for task allocation and job scheduling have been successfully used in several industrial applications. In the previous chapter we already discussed the impressive performance of a joint ACO and GA algorithm for optimal scheduled deliveries of liquid gas to 8000 customers at American Air Liquide. The cost savings and operational efficiencies of this application for one of their 40 plants are more than \$6 million dollars per year.

  Another application of swarm intelligence-based scheduling of truck painting at General Motors claims at least \$3 million per annum. An interesting industrial scheduling problem in an Alcan aluminum casting center was successfully resolved by using ACO which generates 60 optimal schedules in less than 40 seconds.[19]
- *Routing* – One of the first successful ACO application was finding optimal routs for Southwest Airlines cargo operations. The derived solutions looked strange, since it was suggested to leave cargo on a plane headed initially in the wrong direction. However, implementing the algorithm resulted in significant cutback on cargo storage facilities and reduced wage costs. The estimated annual gain is more than \$10 million.[20]

[19]M. Gravel, W. Price, and C. Cagne, Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic, *European Journal of Operating Research*, *143*, pp. 218–229, 2002.

[20]E. Bonabeau and C. Meyer, Swarm Intelligence: A whole new way to think about business, *Harvard Business Review*, May 2001.

Another fruitful application area is optimal vehicle routing. Several impressive applications have been implemented by the Swiss company AntOptima.[21] Examples are: DyvOil, for the management and optimization of heating oil distribution; OptiMilk, for improving the milk supply process, and AntRoute, for routing of hundreds of vehicles of main supermarket chains in Switzerland and Italy.

- *Telecommunication Networks* – Optimizing telecommunication network traffic is a special case of routing with tremendous value creation potential due to the large volume. This is a difficult optimization problem because traffic load and network topology vary with time in unpredictable ways and the lack of central coordination. All of these features suggest that ACO could be a proper solution for this type of problem. A special routing algorithm, called AntNet, has been developed and tested on different networks under traffic patterns. It proved to be very robust and in most cases better than the competitive solutions.[22] ACO has been used by leading telecommunication companies like France Telecom, British Telecom, and the former MCI WorldCom.
- *Process Optimization* – Recently PSOs have been applied in several process optimization problems. Some applications in The Dow Chemical Company include using PSO for optimal color matching, foam acoustic optimal parameter estimation, crystallization kinetics optimal parameter estimation, and optimal neural network structure selection for day-ahead forecasting of electricity prices.[23] Examples of other interesting applications in this area include numerically controlled milling optimization, reactive power and voltage control, battery pack state-of-charge estimation, and cracking furnace optimization.

  A very broad potential application area is using PSO for optimizing data derived from statistical design of experiments. A PSO application for ingredient mix optimization in a major pharmaceutical corporation demonstrated that the fitness of the PSO-derived optimal solution is over twice the fitness found by the statistical design of experiments.[24]
- *Military* – The most well-known military application of swarm intelligence is developing a "swarm" of small unmanned aerial vehicles (UAV) with the capabilities to carry out key reconnaissance and other missions at low cost. For example, a swarm of surveillance UAVs could keep watch over a convoy, taking turns to land on one of the trucks for refueling. Working together as a team, they would ensure complete surveillance of the area around the convoy. Other applications include indoor surveillance. In recent tests up to five radio-controlled

[21] www.antoptima.com

[22] M. Dorigo, M. Birattari, and T. Stützle, Ant colony optimization: artificial ants as a computational intelligence technique, *IEEE Computational Intelligence Magazine*, *1*, pp. 28–39, 2006.

[23] A. Kalos, Automated neural network structure determination via discrete particle swarm optimization (for nonlinear time series models), *Proc. 5th WSEAS International Conference on Simulation, Modeling and Optimization*, Corfu, Greece, 2005.

[24] J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, 2001.

helicopters are being used to collaboratively track small ground vehicles and land on the back of small moving platforms.

A different approach is the "cooperative hunters" concept, where a swarm of UAVs are searching after one or more "smart targets", moving in a predefined area while trying to avoid detection. By arranging themselves into an efficient flight configuration, the UAVs optimize their combined sensing and are thus capable of searching larger territories than a group of uncooperative UAVs. Swarm control algorithms can optimize flying patterns over familiar terrain and introduce fault tolerance to improve coverage of unfamiliar and difficult terrain.[25]

- *Medical* – One of the first PSO medical applications is for successful classification of human tremors, related to Parkinson's disease (see pp. 382–389 in the reference in Footnote 24). A hybrid clustering approach based on self-organizing maps and PSO was applied in different cases for gene clustering of microarrays. Recently, the idea of using a swarm of nanoparticles to fight cancer cells has come close to reality. A research team led by scientists at The University of Texas M.D. Anderson Cancer Center and Rice University has shown in preclinical experiments that cancer cells treated with carbon nanotubes can be destroyed by noninvasive radio waves that heat up the nanotubes while sparing untreated tissue. The technique completely destroyed liver cancer tumors in rabbits without side effects.[26]

6.6 Swarm Intelligence Marketing

As in the case of evolutionary computation, the generic concept of swarm intelligence is easy to explain. However, one potential source of confusion could be the existence of two different approaches – ACO and PSO. It is recommended to emphasize and demonstrate the common basis of both methods – social interaction. An example of a swarm intelligence marketing slide, organized in this philosophy, is given in Fig. 6.21.

The motto of swarm intelligence: "Transfer social interactions into value" captures the essence of the value creation basis of the approach. The left section of the marketing slide represents the generic view of the approach and presents the two key methods (ACO and PSO), inspired by ants and bird flocks. It is focused on the three key phases of swarm intelligence: (1) analyses of social interactions in biology (represented by ants and birds); (2) derived algorithms for optimization; and (3) the results of finding the optimal solution.

---

[25] Smart Weapons for UAVs, Defense Update, January 2007.

[26] C. Gannon, *et al.*, Carbon nanotube-enhanced thermal destruction of cancer cells in a noninvasive radiofrequency field, *Cancer*, *110*, pp. 2654–2665, 2007.

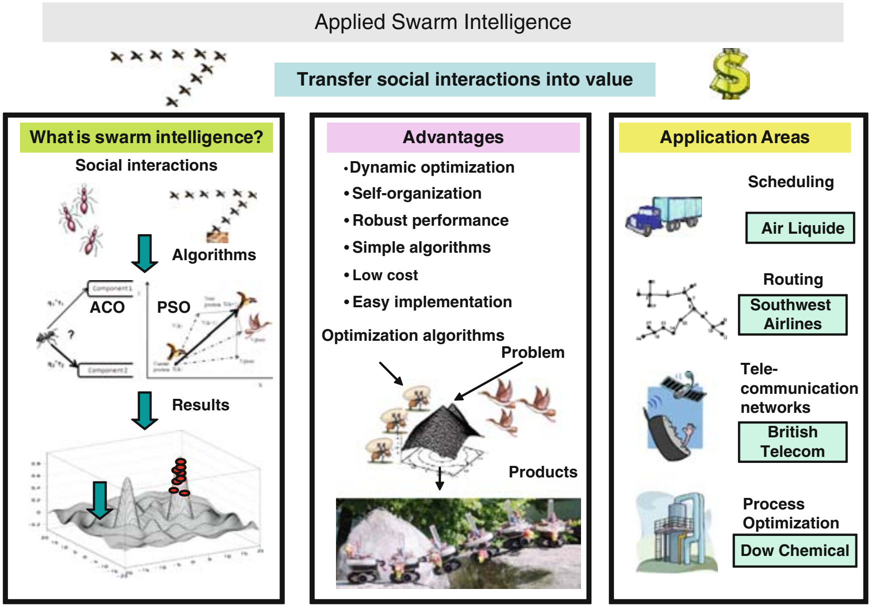


Fig. 6.21 Swarm intelligence marketing slide

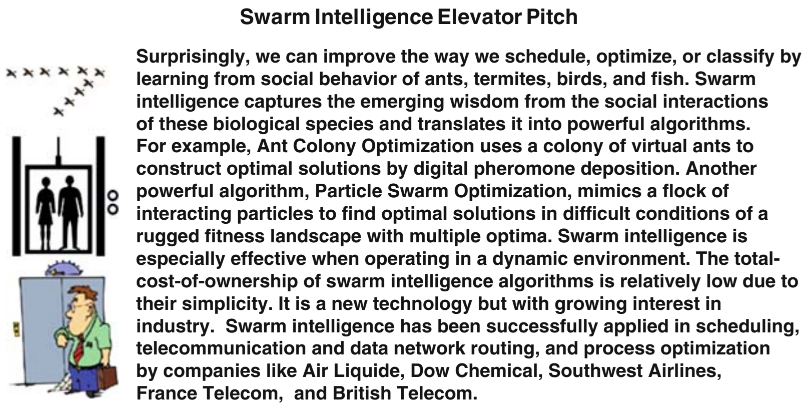


Fig. 6.22 Swarm intelligence elevator pitch

The middle section of the marketing slide represents the key advantages of swarm intelligence, such as derivative-free optimization, self-organization, robust performance, simple algorithms, low cost, and easy implementation. The visualization section represents the problem as a fitness landscape with optima, both optimization algorithms as ants and birds targeting the optimum, and a line of swarming mini-robots, as products.

The key application areas of swarm intelligence are shown in the right section of the marketing slide on Fig. 6.21. The slide includes the most valuable swarm intelligence application areas in scheduling, routing, telecommunication networks, and process optimization. Examples of leading companies in the specific application areas are given.

The proposed elevator pitch for inspiring managers about the great capabilities of swarm intelligence is shown in Fig. 6.22.

6.7 Available Resources for Swarm Intelligence

6.7.1 Key Websites

The key PSO site:
http://www.swarmintelligence.org/

Another PSO site with information and free code:
http://www.particleswarm.info/

The key ACO site:
http://www.aco-metaheuristic.org/

6.7.2 Selected Software

PSO Matlab toolbox (free for noncommercial use)
http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=7506

Several ACO packages, free for noncommercial use are available on the site:
http://www.aco-metaheuristic.org/aco-code/public-software.html

6.8 Summary

Key messages:

Swarm intelligence is coherence without choreography and is based on the emerging collective intelligence of simple artificial individuals.

Swarm intelligence is inspired by the social behavior of ants, bees, termites, wasps, birds, fish, sheep, even humans.

Ant Colony Optimization (ACO) uses a colony of artificial ants to construct optimal solutions for a defined problem by digital pheromone deposition and heuristics.

Particle Swarm Optimization (PSO) uses a flock of communicating artificial particles searching for optimal solutions of a defined problem.

Swarm intelligence-based systems have been successfully applied in scheduling, telecommunication and network routing, process optimization, and different military and medical applications.

The Bottom Line

Applied swarm intelligence has the capability to transfer the algorithms derived from social interaction of artificial individuals into value.

Suggested Reading

The following books give detailed technical descriptions of the different swarm intelligence techniques:

E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence*Swarm Intelligence*: From Natural Evolution to Artificial System*s, Oxford University Press, 1999.

L. de Castro, *Fundamentals of Natural Computing*, Chapman & Hall, 2006.

M. Dorigo and T. Stutzle, *Ant Colony Optimization*Optimization, MIT Press, 2004.

A. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*Swarm Intelligence, Wiley, 2005.

J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, 2001.