Swarm Intelligence
David Bradford and Chih-Cheng Hung

Biometric Recognition Research Center
Southern Polytechnic State University, Marietta, GA 30060-2896 USA

Introduction

Swarm Intelligence. Does that phrase bring to mind the 1978 movie "The Swarm" in which deadly African bees (figure 1) spread through the US killing thousands? Or perhaps it spurs memories of the 1954 movie "Them!" about ants (mutated by atomic testing) that threaten civilization? No? Maybe the phrase swarm intelligence evokes images of insects working together to create works of surprising size and shape – giant termite mounds (figure 2) or other complex nesting structures? Perhaps it brings to mind the ability of a group of insects to behave as if they share some psychic connection that allows them to work together to achieve success over an obstacle or situation? Whether the term swarm intelligence brings to mind movies or the social interactions of the insects involved, what both images share is the concept of a group of entities achieving more together than they could individually.

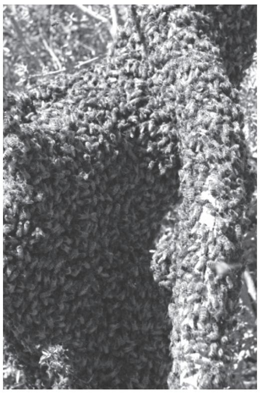


Figure 1 A swarm of bees [Fir0002/Flagstaffotos, 2006].

The successes that nature-based swarm intelligences achieve are only successes when we apply an idea of an obstacle to be overcome – and not just any obstacle but one that would seemingly be beyond the capacity of a single individual of the swarm to grasp or solve. When these individual elements work together to solve a problem - when a group of ants build a living bridge to cross a landscape hindrance (whether water or a gap between two limbs in a tree), when bees harvest pollen from a plentiful source instead of wasting time pursuing poor sources, then their resulting solution benefits the whole swarm. Since nature-based swarms are so successful, can we understand and achieve something new, something also beneficial, about problem solving by imitating them? Not imitating them in a broad sense, but instead in a direct way, by encoding their actions and behaviors and using that encoding to solve problems – and not just generic problems but problems that could have many variables and many conditions that might affect any eventual best solution or solutions.

Figure 2 A termite mound [Yap, 2005].

Computer scientists with the invaluable collaboration of scientists in a multitude of different disciplines have been, in recent years, creating models that are inspired by these natural swarms. They are modeling the behaviors and actions carried out by natural swarms using algorithms. That's right, using algorithms as in figure 3; a series of steps to accomplish a task – but not just a single point a to point b algorithm, like washing hair (wet hair, apply shampoo, lather, rinse, repeat), but algorithms that have many variables and many conditions whose solutions may affect a variety of actions; something like putting together a travel route between lots of cities where there are so many cities to visit and only a certain ordering of visits makes cost-effective sense or monitoring a nuclear reactor and adjusting the energy output to meet the energy demands of the moment without waste or damage – such tasks whose hands-on solutions can take quite a bit of time to calculate.

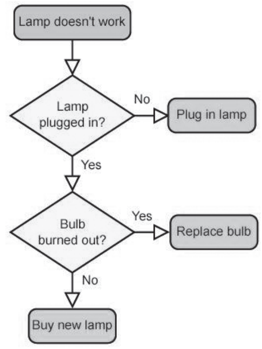


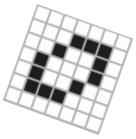
Figure 3 A simple algorithm [BrokenSegue, 2007].

Natural swarms, in an intelligent way that seems beyond the means of its individuals, appear to be able to solve these types of issues quickly and efficiently (finding a short path through a circuitous route to a food supply, or monitoring the population, temperature, honeycomb size and movements – analogies to the earlier presented problems). Computer scientists are now modeling algorithms inspired by natural swarms that get as close as possible to producing the best attainable solutions to solve these types of difficult tasks in an efficient and timely manner! Imitation by algorithmic encoding of natural swarm behavior creates/calculates solutions that appear to have been thought-up by the algorithmic swarm and to have been agreed upon by its members as the solution best suitable to the problem at hand; thus the idea of swarm intelligence in algorithms.

Pioneering Work

Being by no means an exhaustive listing of every scientist involved in swarm intelligence research and development, the following researchers and cited works are highlights that are used as stepping-stones to illustrate, broadly, the emergence and investigation of swarm based algorithms, from past to present (2009).

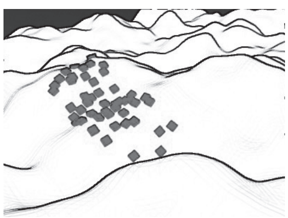
1970



In the year 1970 John Conway created "the best know example of cellular automaton" [Wikipedia, Conway's Game of Life[1] with the Game of Life. A cellular automaton,[2] it uses a grid structure and assigns each grid cell an on or off state (either shaded or clear) based on a number of rules. Conway's Game of Life, with its simple rules and visually appealing representation allowing user interaction "provides an example of emergence[3] and self organization … because of the surprising ways in which patterns can evolve" [Wikipedia, Conway's Game of Life]. While cellular automaton are not specifically a swarm intelligence, the methods used and the 'seemingly intelligent' evolution of some patterns speaks to the general history leading toward swarm intelligence studies.

[1] http://www.bitstorm.org/gameoflife/
[2] Pronounced 'ought-TOM-ah-tawn', meaning self-operating, moving automatically without outside interference.
[3] Emergence is the way complex systems and patterns arise out of a multiplicity of relatively simple interactions. [Wikipedia, Emergence]

1986



In 1986 Craig Reynolds created "a computer model of coordinated animal motion such as bird flocks and fish schools" [Reynolds] which he called Boids. His creation was similar to, but different from, cellular automatons. His Boids (represented in three dimensions instead of using a two dimensional grid-based visualization) also follow simple rules. These animated specks, as seen on his website: http://www.red3d.com/cwr/boids/, can be said to be a swarm implementation of individuals which exhibit many of the same coordinated movements of the life forms they represent (namely birds or fish). In the same way that John Conway's Game of Life displays examples of pattern formation and self organization, so too do Craig Reynolds' Boids display, through the calculation of simple rules, apparent self organization and intelligent tendencies of crash avoidance.

1989

Gerardo Beni with Jin Wang, in 1989, first coined the term "swarm intelligence" when working together in 1989 on cellular robotics [Wikipedia, Gerardo_Beni]. In their article "Swarm Intelligence in Cellular Robotic Systems" they write (about swarm intelligence) "systems of non-intelligent robots exhibiting collectively intelligent behavior evident in the ability to unpredictably produce 'specific' (i.e. not in a statistical sense) ordered patterns of matter in the external environment" [Beni & Wang, 1989]. This definition implies that Conway and Reynolds, in creating their examples through code that self organize and create patterns unpredictably, were in fact executing an early form of swarm intelligence; however, these swarms were not designed as solution seeking algorithms for any array of given problems – not directly.

So, in these early years, these first few pioneers led the way in conceptualizing and defining an idea of swarms of individuals that do something intelligent so that discernable, sometimes unpredictable, patterns emerge.

J. M. Bishop, also in 1989, first elaborated Stochastic[4] Diffusion searching in his paper "Stochastic Searching Networks." He elaborates a method of pattern-matching using cells having randomized values that, when successfully mapped to a solution then diffuses their successful solution to those mapped cells that have invalid values. This may be considered the first of the swarm intelligent algorithms as the individual values, the ants or bees if you will, which are considered active (aka successful) must be communicated to the other solutions which are considered inactive (that do not yet have a successful mapping to a valid solution). Without being a literal interpretation of any particular nature-inspired swarm, this swarm of solutions achieves a final outcome quickly because of sharing successfully mapped values/information - (in the case of Mr. Bishop's paper, a matched pattern of numbers is found).

[4] Pronounced 'stow-KAS-tick', meaning random.

1992

The first true nature-inspired swarm algorithm can be claimed by Ant Colony Optimization and was introduced by Marco Dorigo's Phd thesis "Optimization, Learning and Natural Algorithms" (in the year 1992). It was proposed "as a multi-agent approach to difficult combinatorial optimization problems like the traveling salesman problem[5] and the quadratic assignment problem[6] [Dorigo, Di Caro, & Gambardella, 1999]". In this algorithm the ants solve portions of a given problem then assemble only the best solution portions by combining the ones having the strongest pheromone trail[7]. The swarm in this case communicates its best solutions by attracting more ants toward the stronger pheromone trail (implicit is the abandonment of those trails/solutions that are less reinforced) so that convergence[8] toward the best possible total solution, for example the shortest route to a food source, can happen quickly. This quick convergence illustrates the calculation of an answer from numbers of individuals who by themselves do not calculate the answer as a whole. Instead each is contributing their solution portion and producing an eventual swarm-determined solution thus giving the idea that the swarm as a whole is exhibiting an intelligence that is greater than that of any individual.

[5] Given a list of cities and their pairwise distances, the task is to find a shortest possible tour that visits each city exactly once. [Wikipedia, Traveling Salesman Problem]
[6] A set of n activities/items must be assigned to n locations/resources in such a way that a cost function of the couplings is minimized.
[7] The pheromone trail is a chemical deposit left by ants that is 'attractive' to others ants but which has a limited 'lifespan' and evaporates if not reinforced by that same or other ants as they travel.
[8] In this context, convergence is the gathering of individuals of a swarm around a certain solution value (or values).

1995

James Kennedy (a social psychologist) and Russell Eberhart (an electrical engineer), in 1995, produced a paper "Particle Swarm Optimization" in which they proposed the particle swarm optimization algorithm. By blending their two disciplines they conceived of an algorithm that would have an individual solution (an individuals' solution which would become the best overall solution if conditions warranted) and a group best solution (a best overall solution). The individuals would search for solutions between their best solution and the groups' (termed a swarm) best solution so that any solution found by one individual that was the best solution would attract the other individuals to search in that 'best' solution area. This would only be possible if there were a number of individuals, termed particles, who communicated amongst themselves their solutions so that a best overall solution/direction could be pursued.

This particle swarm, while not directly taking inspiration from insects in the way that Ant Colony Optimization does, can be deemed a fully connected social-entity where individuals determine their own solutions to the problem and then agree upon a solution as found by the whole swarm as their best solution. In this way the swarm moves quickly through the problem landscape toward the best-discovered solution, calculating a solution that all particles of the swarm seem to have also found, thus giving the appearance that they intelligently converged upon a commonly agreed upon solution.

2005

In 2005 at Cardiff University, "researchers at the Manufacturing Engineering Center (MEC) developed the" [Caridff University] Bees Algorithm. While the algorithm itself is of the year 2005, the interpretation of bees as a swarm system was mentioned as early as 1999 in the book "Swarm Intelligence: From Natural to Artificial Systems" [Bonabeau, Theraulaz, & Dorigo, 1999]. Based on the waggle-dance of honeybees, this algorithm exploits the communication between foraging members of a swarm to glean information about the location and quantity of pollen (a solution) in order to send an increased number of swarm-mates back to those solutions that are most promising while other members continue their search. In this way the bee swarm quickly approaches the best pollen supply or supplies and can appear to have intelligently found the overall best solution (s).

What is swarm intelligence?

Is swarm intelligence easy to define, does it have one definition? Assuredly not, in both cases. The intelligent behavior seemingly observed in schooling fish or flocking birds is different from that of ants and bees, which is also different from the actual implementation of a swarm intelligent algorithm. Each does, however, share some common characteristics that allow them to be grouped under the umbrella-term of swarm intelligence.

In nature:

The components of any living swarm[9] tend to be self-organizing meaning that they are able to make independent decisions based on their surroundings and behave accordingly to the stimulus they encounter. They don't crash into each other, unless circumstances arise that allow that action, and they cooperate for the greater benefit of their swarm by performing necessary tasks as dictated by their environment and their current role.

[4] Swarm describes a behaviour of an aggregate of animals of similar size and body orientation, often moving en masse in the same direction. [Wikipedia, Swarm]

In this natural sense then the definition of a swarm of insects is also a flock of birds, a shoal of fish, a herd of land animals, a pod of whales – in essence any "collective motion of a large number of self-propelled entities" [Wikipedia, Flocking behavior]. They, the natural swarms, therefore all share these properties: there are many, relatively homogenous, individuals who interact with each other and their environment (using positive and negative reinforcement) to form a stable and cohesive pattern making whole [Scholarpedia] [Liu & Passino, 2000] [Beni, Order by Disordered Action in Swarms] [Krink]. When these natural swarms' properties are viewed as a whole then a definition of swarm intelligence can be formulated; many variations such a definition exist, one being: "A swarm has been defined as a set of (mobile) agents which are liable to communicate directly or indirectly (by acting on their local environment) with each other, and which collectively carry out a distributed problem solving" [Hoffmeyer]. Others definitions, as a sample, are available from [White] and [Google, swarm intelligence].

In algorithms:

The idea of a natural swarm being intelligent, or not, is an externally imposed human-based interpretation applied to the actions and/or results achieved by the individuals of that natural swarm when such are seemingly beyond the capabilities of its members independently. In those cases the intelligence observable is termed to be emergent. This idea is equally applicable to a swarm intelligent algorithm. The intellectual jump from nature-based swarms to an algorithmic counterpart can be seen in the following quote:

> "a social insect colony is undoubtedly a decentralized problem-solving system, comprised of many relatively simple interacting entities." "the modeling of social insects by means of SO (self organization) can help design artificial distributed problem-solving devices that self-organize[10] to solve problems – swarm intelligent systems." [Bonabeau, Theraulaz, & Dorigo, 1999]

Therefore, looking at a natural swarm as 'simple interacting entities that problem solve' is easily re-interpreted to be an algorithm having multiple variables (the swarm) that take-on solution values[11] appropriate to the problem at hand[12]. These variables may compare their values with each other, build a best value found-so-far collection and, without individually knowing the solution, arrive as a whole at a solution that is suitable for the algorithmic swarms' objective (i.e. the problem at hand).

[9] Swarm describes a behaviour of an aggregate of animals of similar size and body orientation, often moving en masse in the same direction. [Wikipedia, Swarm]
[10] Self-organization is a process of attraction and repulsion in which the internal organization of a system, normally an open system, increases in complexity without being guided or managed by an outside source. Self-organizing systems typically (but not always) display emergent properties. [Google, self organization]
[11] To 'take on solution values' is to perform calculation (s) on the variable value and evaluate the value arrived at for appropriateness as a valid possible answer to the objective problem being solved. This process is repeated for each 'entity of the swarm' for as long as the swarm is tasked to solve the problem (the swarms' lifespan).
[12] Where 'the problem at hand' is the problem landscape (that range of valid solutions) defined by the objective the swarm is assigned to achieve/solve.

So how can all those earlier stated properties of a natural swarm be interpreted so that they may be used as building blocks of a swarm-based algorithm? Here are 5 principles for algorithmic swarms:

1. Proximity – The population should be able to carry out simple space and time computations.
2. Quality – The population should be able to respond to quality factors in the environment.
3. Diverse Response – The population should not commit its activity along excessively narrow channels.
4. Stability – The population should not change its mode of behavior every time the environment changes.
5. Adaptability – the population must be able to change behavior mode when it's worth the computational price.
   (points 1-5 as quoted from Mark Millonas by [Eberhart, Shi, & Kennedy, 2001])

A swarm algorithm's intelligence is directly proportional to its ability to produce patterns of solutions to a problem that can be understood as emergent, that is, understood/perceived by an entity external to the swarm as a collaboration of values whose calculated solutions, achieved without explicit direction from any single entity of the algorithmic swarm, are presented as a single collective whole. Normally, a swarm algorithm takes only one facet of a swarm's activity and interprets that facet – e.g. bringing food back to the nest. A swarm intelligent algorithm does not try to also focus on building of the nest or storage of food in the nest nor any other division of labor of the entities. Because of this single-facet-focus a swarm algorithm may have a pared-down architecture (a more restrictive set of rules for use) that could make the emergence of intelligence in its pursuit of a solution less obvious. However, this focused approach allows for a swarm algorithm to be encoded in such a way as to return potentially more reliable results that occur in the range expected.

Types of swarm intelligence

Ant Colony Optimization (ACO)

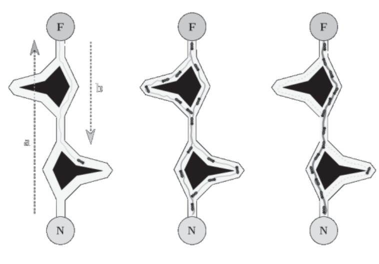


Figure 4 Finding the shortest path using pheromone reinforcement for the shortest path [Nojhan, 2006].

The image in Figure 4 is one of the classic interpretations of how an ant colony quickly finds the shortest route between a food source (F) and their nest (N). Stigmergy, the reinforcement of a pheromone trail, is the active responsible agent which entices the ants to remain on the shortest path and abandon the longer routes (since the enticing pheromones evaporate quickly, the longer routes are not attractive).

"The ant colony optimization algorithm is a probabilistic[13] technique for solving computational problems which can be reduced to finding good paths through graphs" [Answers.com website]. The algorithm works with nodes, aka locations, that an ant can travel to that have an associated 'cost' - travel to a good node may have a low cost where to a bad node may have a high cost. As the ants travel from node to node (and not all ants must travel to all possible nodes) those node-connections that incur the least cost are termed the optimal, or best, paths. In this way the shortest paths that have been traveled by any ant can be assembled to represent the single shortest path, and without any one ant necessarily traveling the entire route from starting node to ending node. The emergent intelligence of the swarm solution is the shortest overall path/solution attained by its members.

In setting up the algorithm, a programmer would encode the number of ants and number of nodes, the costs associated to reach each and any other node specific information, including any constraints involved (e.g. node 5 can only be reached from node 3), the amount of attractive pheromone that an ant should deposit (to lure other ants toward its better solution) and the evaporation rate of that pheromone, and an algorithm termination condition. (paraphrased/compiled from: [Dorigo & Di Caro, 1999])

[13] Any algorithm that works for all practical purposes but has a theoretical chance of being wrong. [NIST, probabilistic algorithm]

Bee Algorithms

The Bees Algorithm is an optimization algorithm inspired by the natural foraging behavior of honey bees to find the optimal solution, i.e. most plentiful food source. The algorithm requires a number of parameters to be set, namely: number of scout bees (n), number of sites selected out of n visited sites (m), number of best sites out of m selected sites (e), number of bees recruited for best e sites (nep), number of bees recruited for the other (m-e) selected sites (nsp), initial size of patches (ngh) which includes site and its neighborhood and stopping criterion. [Answers, Bees-algorithm] / [Wikipedia, Bees-algorithm]

The emergent intelligence involved is the ability of the swarm to locate a superior solution source and quickly optimize that source; should the source become exhausted (or the best solution source 'change') then the algorithm can adapt readily.

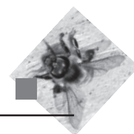
Particle Swarm Optimization (PSO)

PSO is best described by:

"A problem is given, and some way to evaluate a proposed solution to it exists in the form of a fitness function. A communication structure or social network is also defined, assigning neighbors for each individual to interact with. Then a population of individuals defined as random guesses at the problem solutions is initialized. These individuals are candidate solutions. They are also known as the particles, hence the name particle swarm. An iterative process to improve these candidate solutions is set in motion. The particles iteratively evaluate the fitness of the candidate solutions and remember the location where they had their best success. The individual's best solution is called the particle best or the local best. Each particle makes this information available to their neighbors. They are also able to see where their neighbors have had success. Movements through the search space are guided by these successes, with the population usually converging, by the end of a trial, on a problem solution better than that of non-swarm approach using the same methods." [Answers, particle-swarm-optimization]

PSO's emergent intelligence is its ability to take an objective function and, without any one particle defined as a permanent leader, arrive as a collective swarm at a solution fitting the objective. In setting up the algorithm a programmer would encode the number of particles, the termination condition, the objective function whose calculation results in a fitness, or cost, value of each solution so that a determination can be made as to that solution being better than the particle or swarms solution so far, and the particles communication method (via neighbors or through a single particle or all particles).

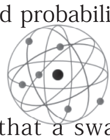
Swarm intelligence and traditional optimization tradeoff



You might wonder what you gain by using a swarm intelligent algorithm, or, on the flip side of the coin, what you lose. Well, the gains and losses aren't easy to measure since any win against a particular problem is possibly not a win against another problem. What we mean is that use of a swarm intelligent algorithm should be properly weighed against the type of problems you are hoping that algorithm will solve as well as any necessity for speed (to find any acceptable solution quickly) versus absolute certainty of perfect, precise, solution.

"Speed and precision are conflicting objectives, at least

in terms of probabilistic algorithms” [Weiss] . This quote illustrates one of the core issues for algorithms – do you want a blindingly fast algorithm that can answer the given problem pretty much successfully the majority of the time, or, do you want an answer to the problem that is as precise and perfect as can be achieved regardless of the time involved? A perfect environment would allow blindingly fast, perfect and precise answers to a problem, but with optimization algorithms[14] this is not yet the case. “Generally, optimization algorithms can be divided in two basic classes: deterministic[15] and probabilistic algorithms” [Weiss] .



Simply by their nature, by the very fact that a swarm intelligent algorithm is comprised of many individuals, and those individuals have a certain ‘freedom of calculation’ in the solutions they generate, we can reasonably say that a swarm intelligent algorithm could easily fall into the class of probabilistic algorithms. The swarm, as a whole, could evaluate the problem and arrive at an answer that is not the one true answer every single time; that possibility does exist – however, precisely because a swarm does have many individuals pursuing solutions, there is always the high probability that one or more individuals will absolutely find (if not the one true answer) an answer that is exceedingly well suited to the problem, termed an optimal solution. “One of the most fundamental principles in our world is the search for an optimal state .” [Weiss]

“…no one optimization algorithm can possibly be efficient or even successful in all cases of interest” [Coool] . So, what is the trade off using swarm intelligent algorithms? On the one hand you get, traditionally, faster solutions to problems that would take an inordinate amount of time to evaluate. On the other hand you lose the ability to say that an answer to a problem will always, definitively, be the best answer and may in fact, once in a blue moon, be an answer that isn’t that good at all! This is where your knowledge of the problem becomes of great importance – is it necessary for your purposes (and the problem being considered) to have a penultimate optimal solution or can you work with and use an answer that may be very, very good?

Consider this simple example problem of driving down the road in a car and having to stay between a pair of lines painted on the road, termed a lane – it is not paramount that the distance to the left of the car and the left line be exactly the same as the distance between the right side of the car and the right line; the answer of staying between the lines and in the lane allows for any distance on either side of the lines to their corresponding side of the car to be OK as long as the car does not ‘cross’ either of the lines – therefore the optimum answer is a range of answers, really, and not a single position within the demarcated lane. A swarm intelligent algorithm would be an excellent choice to solve this type of problem since the lane size/width could change at any moment (and the algorithm’s individuals, the solutions, could adapt to that change easily and quite quickly) .

[14] An optimization algorithm is a numerical method or algorithm for finding a value x such that f (x) is as small (or as large) as possible, for a given function f, possibly with some constraints on x. [Wikipedia, list of optimization algorithms]
[15] An algorithm whose behavior can be completely predicted from the input. [NIST, deterministice algorithm]

Applications of swarm intelligence algorithms

“Swarm intelligence has applications in decentralized controls of unmanned vehicles for the military so single operators can control more unmanned vehicles. The use of swarm intelligence in medical nanobots may also help combat cancer. Swarm intelligence was used in the creation of the video sequence "Battle of Helm's Deep" in the movie, Lord of the Rings.” [TechFaq]

“Nature-inspired approaches have not only shown their efficiency in static optimization problems, but were proven to be especially robust in dynamic applications, too. This is particularly interesting in the looming age of networks of larger scale. Wireless networks, sensor networks, wireless sensor networks, Smart Home networks, ubiquitous computing, and more require self-organization, efficient routing, optimal parameter settings, and power management.” [Weiss]

The above quotes are used to illustrate the hallmarks of problems that could be effectively addressed by a swarm intelligent algorithm. They, the problems, use or represent many individuals that are all part of a common task. They work over a range of solutions that are all ‘optimally’ valid. They must and do take into account various constraints upon their solutions so that evaluations/calculations do not lead the swarm out of the desired solution landscape. They are in some respects time sensitive, requiring an acceptable answer in minimal time from a problem that may be dynamic[17] in nature.

There are also many academic and research implementations of swarm intelligent algorithms – and many hybridizations and modifications that have been investigated for purposes specific to the researchers needs.

[17] In a state of flux, or, having changes to its requirements on an ongoing basis.

Conclusion

Swarm intelligence, as applied to algorithms, isn't such a mystery anymore, now is it? We've seen that a natural swarm isn't anything more than a conglomeration of entities and proposed that intelligence in a natural swarm is only our own ability to recognize patterns whose creation seemingly are beyond the comprehension or direction of a single individual of that swarm. We've conducted a brief survey through the history of swarm intelligence computation, selecting highlights from those we've termed pioneers, which illustrate how swarm intelligent algorithms may have evolved in our understanding and estimation. We further discussed some of the hallmarks of swarm behavior (something akin to the rules of swarm interaction) and presented some of the manners and methods those behaviors and properties have been transferred/reinterpreted into a selection of swarm based intelligent algorithms. We then delved briefly into the optimization issues presented swarm intelligent algorithms, highlighting strength (s) and weakness (es) for solving types of problems, and followed that with some example situations in which swarm intelligent algorithms might be and have been successfully employed.

As researchers continue to explore what swarm intelligence is and is not, and that understanding gets imported and applied to optimization problems through swarm intelligent algorithms, more and more hybridizations will emerge. Whether these imported modifications and hybridizations will remain termed swarm intelligent algorithms is up to the researchers and both their goals and interpretations of their end product/algorithm. It stands to reason that if an algorithm maintains or produces a group of probable solutions to a problem from which a solution or set of solutions is mutually agreed upon by the group as 'best' then that algorithm is partaking in swarm intelligence to some degree – even if no discernable pattern emerges (our ability to pick out a pattern is not guaranteed; a pattern may be present and not yet ascertainable or discovered by our inquiries).

Solutions to problems that emerge from a grouping of individuals, purposefully or by unintentional group synergy, allow a natural swarm to be resilient, to survive in the face of difficulties that may not be known – this type of flexibility of the group (to respond to the unknown) allows for an interpretation of intelligence as the group creates solution-patterns that it couldn't have planned for previously. This too is the goal of swarm intelligence in algorithms – to solve problems whose solution-patterns haven't before been formed and couldn't be achieved by the individuals of the swarm alone, but, when working together as a group create high quality solutions that emerge and appear inevitable.

Whether the term swarm intelligence first brings to mind the movies or cooperative insect societies, we hope this brief overview of its algorithmic-interpretation has added depth to your understanding of an interesting and innovative method of problem solving.

Appendix A：

"some of the practical issues that arise in attempting to find the appropriate tool (i.e. algorithm) for a given problem" [Coool] – all bullets：

"some of the practical issues that arise in attempting to find the appropriate tool (i.e. algorithm) for a given problem" [Coool] – all bullets：

•**Unimodal functions.** These are functions which have a single extremum. The archetype of such functions is the dimensional quadratic form. Non-quadratic, but still unimodal, functions can usually be optimized by making a sequence of quadratic approximations. If the matrix of second derivatives of a quadratic form is known, then special-purpose algorithms can be used.

•**Essentially unimodal functions.** Because of noise and other factors, it is often the case in practice that the simple global structure of a problem is masked by parasitic local optima. We do not regard these local optima as representing important features of the model and so, if we could somehow smooth the objective function, we would be more confident that we had determined the extremum we seek.

•**Functions that have a small number of significant local optima.** Sometimes the local optima represent significant features in the model (for example, fundamental ambiguities can exist in inverse calculations; the global optimum may not be fundamentally more significant than other local optima). In this case we should not simply smooth the objective function; we must find these local optima. If their number is relatively small, then we might be able to rely on hill climbing from randomly chosen starting points.

•**Functions with significant null-space effects.** In the event that the objective function becomes flat in the neighborhood of the current point, then this flat region represents perturbations to the model which have little or no influence on the objective function. In this case it is important to map out these regions and characterize the ambiguities that they represent.

•**Functions with a huge number of significant local optima.** It may happen that there are a large number of local optima, many of which are significant. We cannot ignore them by smoothing the objective functions, and it may be that random hill climbing is too inefficient. This is where Monte Carlo methods such as Simulated Annealing and Genetic Algorithms are normally used.

•**Functions whose global structure provides no useful information.** If the objective function is essentially flat, except for an isolated, deep optimum, then the global structure of the function is of no use in finding the desired model. Unless an alternative parameterization can be found in which the function has some global structure, such as in B, extensive brute force random searching or enumeration may be the only alternative.

Bibliography

Answers. Bees-algorithm. Retrieved 12 2009, from Answers.com: http://www.answers.com/topic/bees-algorithm

Answers. particle-swarm-optimization. Retrieved 12 2009, from Answers.com: http://www.answers.com/topic/particle-swarm-optimization

Answers.com website. Ant Colony Optimization. Retrieved 12 2009, from Answers.com: http://www.answers.com/topic/ant-colony-optimization

Beni, G. Order by Disordered Action in Swarms. Retrieved 12 2009, from swarm-robotics.org: www.swarm-robotics.org/SAB04/presentations/beni-research.ppt

Beni, G., & Wang, J.（1989）. Swarm Intelligence in Cellular Robotic Systems. Proceedings of NATO Advanced Workshop on Robots and Biological Systems , 102.

Bishop, J. M.（1989）. Stochastic searching networks. First IEE International Conference on Artificial Neural Networks（pp. 329-331）. London: IEEE.

Bonabeau, E., Theraulaz, G., & Dorigo, M.（1999）. Swarm Intelligence. Oxford University Press.

BrokenSegue. Lamp Flow Chart.

Caridff University. The Bees Algorithm. Retrieved 12 2009, from Bee Algorithm: http://www.bees-algorithm.com/modules/1/0.php

Coool. Why we need lots of optimization algorithms. Retrieved 12 2009, from coool.mines.edu: http://coool.mines.edu/report/node3.html

Dorigo, M., & Di Caro, G.（1999）. 2, pp. 1470-1477. Washington DC: IEEE.

Dorigo, M., Di Caro, G., & Gambardella, L. M.（1999）. Ant algorithms for discrete optimization. Artificial Life , 5（2）, 137-172.

Eberhart, R. C., Shi, Y., & Kennedy, J.（2001）. Swarm Intelligence. Morgan Kaufmann.

Fir0002/Flagstaffotos. Bee Swarm On Fallen Tree. http://commons.wikimedia.org/wiki/File:Bee_swarm_on_fallen_tree02.jpg.

Google.（n.d.）. self organization. Retrieved 12 2009, from Wikipedia.org: http://www.google.com/search?q = define% 3Aself-organization&sourceid = ie8-activity&hl = en

Google.（n.d.）. swarm intelligence. Retrieved 12 2009, from Wikipedia.org: http://www.google.com/search?hl = en&defl = en&q = define:Swarm+intelligence&ei = RyQsS76sFsz_nAfb6pyCCQ&sa = X&oi = glossary_definition&ct = title&ved = 0CAcQkAE

Hoffmeyer, J. Swarm-semiotics. Retrieved 12 2009, from Molbio.ku.dk: http://www.molbio.ku.dk/MolBioPages/abk/PersonalPages/Jesper/Swarm.html

Krink, T. Swarm Intelligence - Introduction. Retrieved 12 2009, from staff.washington.edu: http://staff.washington.edu/paymana/swarm/krink_01.pdf

Liu, Y., & Passino, K. M.（2000）. Swarm Intelligence: Literature Overview.

NIST. deterministice algorithm. Retrieved 12 2009, from itl.nist.gov: http://www.itl.nist.gov/div897/sqg/dads/HTML/deterministicAlgorithm.html

NIST. probabilistic algorithm. Retrieved 12 2009, from itl.nist.gov: http://www.itl.nist.gov/div897/sqg/dads/HTML/probablAlgo.html

Nojhan. ACO Branches.

Reynolds, C. Boids. Retrieved 12 2009, from red3d.com: http://www.red3d.com/cwr/boids/

Scholarpedia. Swarm Intelligence. Retrieved 12 2009, from Scholarpedia.org: http://www.scholarpedia.org/article/Swarm_intelligence

TechFaq. swarm-intelligence. Retrieved 12 2009, from tech-faq.com: http://www.tech-faq.com/swarm-intelligence.shtml

Weiss, T. Global Optimization Algorithms - Theory and Application. Retrieved 12 2009, from it-weise.de: http://www.it-weise.de/projects/book.pdf

White, T. Swarm Intelligence. Retrieved 12 2009, from sce.carleton.ca: http://www.sce.carleton.ca/netmanage/tony/swarm.html

Wikipedia. Bees-algorithm. Retrieved 12 2009, from Wikipedia.org: http://en.wikipedia.org/wiki/Bees_algorithm

Wikipedia. Conway's Game of Life. Retrieved 12 2009, from Wikipedia.org: http://en.wikipedia.org/wiki/Conway% 27s_Game_of_Life

Wikipedia. Emergence. Retrieved 12 2009, from Wikipedia.org: http://en.wikipedia.org/wiki/Emergence

Wikipedia. Flocking behavior. Retrieved 12 2009, from Wikipedia.org: http://en.wikipedia.org/wiki/Flocking_（behavior）

Wikipedia. Gerardo_Beni. Retrieved 12 2009, from Wikipedia.org: http://en.wikipedia.org/wiki/Gerardo_Beni

Wikipedia. list of optimization algorithms. Retrieved 12 2009, from Wikipedia.org: http://en.wikipedia.org/wiki/List_of_optimization_algorithms

Wikipedia. Swarm. Retrieved 12 2009, from Wikipedia.org: http://en.wikipedia.org/wiki/Swarm

Wikipedia. Traveling Salesman Problem. Retrieved 12 2009, from Wikipedia.org: http://en.wikipedia.org/wiki/Travelling_salesman_problem

Yap, B. V. Termite Cathedral. http://commons.wikimedia.org/wiki/File:Termite_Cathedral_DSC03570.jpg.

David Bradford Jr.

received his B.S. in Fine Arts from Auburn University, AL in 1992 and his M.S. degree from Southern Polytechnic State University in Computer Science in 2010. He is currently working on Business Intelligence projects in software applications. His research interests are swarm intelligence, pattern recognition, and image processing.

Chih-Cheng Hung

received his B.S. in business mathematics from Soochow University in Taiwan, and his M.S. and Ph.D. in Computer Science from the University of Alabama in Huntsville, AL in 1986 and 1990, respectively. He is a professor of Computer Science at Southern Polytechnic State University, GA. He developed image processing software tools with the Department of Image Processing Applications at Intergraph Corporation from 1990 to 1993 and an associate professor at Alabama A&M University from 1993 to 1999. He served as the program co-chair for the Association of Computing Machinery (ACM) Symposium on Applied Computing (SAC 2010) – Sierre, Switzerland from March 22 to 26, 2010. He is an associate editor of Information: An Intenational Interdisciplinary Journal. His research interests include image processing, pattern recognition, neural networks, genetic algorithms, artificial intelligence and software metrics.