IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 7, Issue 1, January 2020

Towards Study an Ant Colony Optimization

Manoj Kumar Bisht

Department of Computer Science, Kumaun University, S.S.J. Campus Almora (Uttarakhand), India

Abstract: Ant Colony Optimization (ACO) is a paradigm for designing met heuristic algorithms for combinatorial optimization problems. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions. Heuristic algorithms are algorithms which, in order to escape from local optima, drive some basic heuristic: either a constructive heuristic starting from a null solution and adding elements to build a good complete one, or a local search heuristic starting from a complete solution and iteratively modifying some of its elements in order to achieve a better one. The met heuristic part permits the low-level heuristic to obtain solutions better than those it could have achieved alone even if iterated. Usually, the controlling mechanism is achieved either by constraining. The particular way of defining components and associated probabilities is problem-specific, and can be designed in Different ways, facing a trade-off between the specificity of the information used for the conditioning and the number of solutions which need to be constructed before effectively biasing the probability distribution to favor the emergence of good solutions. Different applications have favored either the use of conditioning at the level of decision variables, thus requiring a huge number of iterations before getting a precise distribution, or the computational efficiency, thus using very coarse conditioning information.

Keywords: ACO, Swarm, Neural Networks, Fuzzy System, Swarm Intelligence, IEEE etc

I. INTRODUCTION

Ant Colony Optimization (ACO) is a paradigm for designing met heuristic algorithms for combinatorial optimization problems. The first algorithm which can be classified within this framework was presented in 1991 and, since then, many diverse variants of the basic principle have been reported in the literature. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions. Met heuristic algorithms are algorithms which, in order to escape from local optima, drive some basic heuristic: either a constructive heuristic starting from a null solution and adding elements to build a good complete one, or a local search heuristic starting from a complete solution and iteratively modifying some of its elements in order to achieve a better one. The met heuristic part permits the low-level heuristic to obtain solutions better than those it could have achieved alone even if iterated. Usually, the controlling mechanism is achieved either by constraining or by randomizing then set of local neighbor solutions to consider in local search (as is the case of simulated annealing or tabu search.

The particular way of defining components and associated probabilities is problem-specific, and can be designed in Different ways, facing a trade-off between the specificity of the information used for the conditioning and the number of solutions which need to be constructed before effectively biasing the probability distribution to favor the emergence of good solutions. Different applications have favored either the use of conditioning at the level of decision variables, thus requiring a huge number of iterations before getting a precise distribution, or the computational efficiency, thus using very coarse conditioning information.

1.1 WHAT IS A SWARM?

It refers to a disorganized cluster of moving things, usually insects, moving irregularly, chaotically, somehow staying together even while all of them move in apparently random direction. It is a population of interacting elements that is able to optimize some global objective through collaborative search of a space. Interactions that are relatively local are often emphasized. There is a general stochastic tendency in a swarm for individuals to moves toward a center of mass in the population on critical dimensions, resulting in convergence to an optimum. E.g. swarm of bees, ant colonies, animal herd, bird flock, and fish school. We can learn about social insect and apply this knowledge to the field of intelligent system design. The complexity and sophistication of self-organization is carried out with no clear leader in insets. The modeling of social insects by means of self-organization can help design artificial distributed problem solving devices. These are also known as Swarm Intelligent System.

The interaction among insects can be classified as:

- **Direct Interaction**

These are the direct physical interactions among the insects. E.g. Food/liquid exchanges, visual contact, chemical contact (pheromones).

- **Indirect Interaction**

Individual behavior modifies the environment, which in turn modifies the behavior of other individuals.

1.2 WHAT IS SWARM INTELLIGENCE?

Swarm Intelligence is a property of system of non-intelligence robots exhibiting collectively intelligent behavior.
Swarm intelligence is a branch of Computational Intelligence. It is an AI technique based around the study of collective behavior in decentralized, self-organized systems.
Computational Intelligence is the branch of the study of AI. Its research aims to use learning, adaptive or evolutionary computations to create programs that are, in some real sense, intelligent.

Subjects in Computation Intelligence as defined by IEEE Computation Intelligence Society are:

- Neural Networks
- Fuzzy Systems
- Evolutionary Computation

Evolutionary Algorithm

Swarm Intelligence.

1.2.1 AIM OF SWARM INTELLIGENCE

The aim of Swarm Intelligence is to write computer programs that simulate societies of individuals, each working on a problem and at the same time perceiving the problem solving attempts of its neighbors, and being influenced by those neighbors' success.
Intelligence is an elusive quality, considered as a characteristic of humans. It arises from interactions among individuals. We humans are the most social out of all animals. We live together in families, tribes, cities, nations, behaving and thinking according to the rules and norms of our communities adopting the customs of our fellows.
Even when we are alone, we think about other people. In real social interaction, information is exchanged; individual exchange rules, tips and beliefs about how to process the information. It topically results in a change in the thinking process, not just the contents, of the participants.

Swarm Intelligence uses findings based on the social behavior of swarm animals to develop answer to challenges faced in today's complex business environment.
In multivalent systems, autonomous subroutines perform specialized functions. Agent subroutines may pass information back & forth, but subroutines are not changed as a result of interactions, as people are.

There are four **principles** governing the swarm intelligence. There are:

- **Positive Feedback –** Swarm Intelligence reinforces good solutions present in the system.
- **Negative Feedback -** It remove old or poor solution.
- **Randomness –** Solutions can be tested regardless of perceived quality, which in turn, result in creative and unconventional solutions.
- **Multiple Interactions –** This is the key to building up the best solutions.

1.2.2 MOTIVATION BEHIND SWARM INTELLIGENCE

Ants, Bees, or Termites - all social insects – show impressive collective problem –solving capabilities. Properties associated with their group behavior like self-organization, robustness and flexibility are seen as characteristics that artificial systems should exhibit for optimization, control or task execution. In the last decade, diverse efforts have been made to take social insects as an example and develop algorithms inspired by their strictly self-organized behavior.

- **Robust nature of animal problem-solving: -**
  - Simple creatures exhibit complex behavior.
  - Their behavior is modified by dynamic environment.

- **Emergent behavior observed in: -**
  - Bacteria.
  - Ants.
  - Bees.

 DOI 10.17148/IARJSET.2020.7101

- **Emergent behavior includes: -**
  - Forming bridges.
  - Raiding specific areas for food.
  - Building and protecting nests.
  - Sorting food items.
  - Cooperating in carrying large items.
  - Emigration of a colony.
  - Finding shortest route from nets to food source.

1.2.3 CHARACTERISTICS OF SWARM INTELLIGENCE

1.2.3.1 Flexibility
The colony can adept to a changing environment. So, Swarm Intelligence techniques are flexible and behave according to the changing environment.

1.2.3.2 Robustness
Swarm intelligence techniques always find an optimal solution and deal well with change. Even when one or more individual fail, the group can still perform its tasks.

1.2.3.3 Decentralized
Activities are neither centrally controlled, nor locally supervised.

1.2.3.4 Self-organized
Self-organization is a set of dynamical mechanisms whereby structure appear at the global level of a system from interactions of its lower-level components.

1.3 SWARM INTELLIGENCE SYSTEM:
Swarm Intelligence systems are made up of a population of simple agents interacting locally with one another and with their environment. Local interactions between such agents often lead to the emergence of global behavior. It provides a basis with which it is possible to explore collective (or distribute) problem solving without centralized control or the provision of a global model.

1.3.1 SWARM OPERATION:
All the Swarm Intelligence systems follow a general approach towards problem solving.
The swarm agents perform the following steps to solve the problem.
They

- Arrive at a node,
- Sense the environment,
- Undertake local activity,
- Modify environment,
- Use sensory input to make migration decision.

1.3.2 AREAS OF APPLICATION

- Design decision rules in complex and volatile environments.
- Optimization problems which are unsuitable or unfeasible for analytical or exact approaches nasty multimode functions, NP-hard problems, non-stationary environments hard to model, distributed systems with hidden states, problems with many variables and sources of uncertainty.
- Planning of productions site/distribution network based on the rules of clustering and strong.
- Implementing flexible “buckets brigades” for optimal allocation of resources with in a process.
- Dynamic problems requiring distributed/multi-agent modeling: collective robotics, sensor networks, and telecommunication networks.
- Problems suitable for distributed/multi-agent modeling: military applications (surveillance with UAVs), simulation of large-scale systems (economics, social interactions, biological systems, virtual crowds).
- Entertainment: video games, collective artistic performance.
- Self-healing, self-organization communication networks
- Self-aggregating devices
- Swarm “Urban combat model”
- Swarm-based sensor networks, smart dust

II. SWARM INTELLIGENC

Swarm Intelligence will provide new approaches to solving mainly organizational challenges in business networks as well as for the optimization of flows in technical networks (e.g. IT/TC networks). It is necessary to deploy solutions based on these approaches due to the increasing complexity involved. It will be interesting to pursue the translation of the knowledge already available in the academic environment into real business life. Finding the right match between the different swarm models and the relevant business challenges is one of the critical success factors. But also the development of new services and business models based on SI-approaches seems to provide an interesting research area.

The two most successful SI techniques currently in existence are:

- **Practical Swarm Optimization (PSO)**
- **Ant Colony optimization (ACO)**

2.1 PRACTICAL SWARM OPTIMIZATION (PSO)

It is a population based stochastic optimization technique inspired by the social behavior of bird flocking or fish schooling.

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA. PSO have no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles fly through the problem space by following the current optimum particles.

Each particle keeps tracks of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far (The fitness value is also stored). This value is called pbest. Another " best" value that is tracked by the particle swarm optimizer is the best value. Obtained so far by any particle in the neighbors of the particle. The location is called 'best'. When a particle takes all the population as its topological neighbors. The best value is a global best and is called 'gbest'

The Particle swarm Optimization concept of at each time step, changing the velocity of (accelerating) each particle toward its 'pbest' and 'lbest' locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration towards 'pbest' and 'lbest' locations.

In past several years. PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods.

Another reason that PSO is attractive is that there are few parameters to adjust. One version with slight variations, works well in a wide variety of applications. PSO has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement.

While ants move more or less randomly around their physical world, bird & fishes move in more orderly ways.

PSO utilizes a "Population" of candidate solutions to evolve an optimal or near-optimal solution to a problem. The degree of optimality is measured by a fitness function defined by the user.

2.2 ANT COLONY OPTIMIZATION (ACO)

The behavior of ants has long fascinated scientists. And why not?

These insects have the strength to carry food up to seven times their own body weight, and set up amazingly complex colonies, with social 'castes' in which every member has a role;

In fact, ants are not only fascinating just to. In recent years, computer scientists have been paying great attention to the way in which a colony of ants can solve complex problems; in particular, how it finds the shortest route to food source.

Each insect in a colony seemed to have its own agenda, and yet the group as a whole appeared to be highly organized. This organization was not achieved under supervision but through interaction among individuals. This was most apparent in the way in which ants travel to and from a food source.

Pheromone: It is a chemical discharged by animal's body & influences the behavior of others of the same species when released in air.

Ants form and maintain a line to their food source by laying a trail of pheromone, i.e. chemical to which other members of the same species are very sensitive. They deposit a certain amount of pheromone while walking and each ant prefers to follow a direction rich in pheromone. This enables the ant colony to quickly find the shortest route. The first ants to return should normally be those on the shortest route, so this will be the first to be doubly marked by pheromone (once in each direction). Thus other ants will be more attracted to this route than to longer ones not yet doubly marked, which means it will become even more strongly marked with pheromone.

Soon, therefore, nearly all the ants will choose this route. But what if the ants happened to return from a longer route first, marking it most strongly? Computer simulations show that this problem is solved if that pheromone trails on longer routes.

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 7, Issue 1, January 2020

Studying this uncanny skill has researchers to create software agents capable of solving complex IT problems. Such as rerouting traffic in a busy communications network.

Ant colonies, and more generally social insect societies, are distributed systems that inspire of the simplicity of their individuals, present a highly structured social organization. As a result of this organization, ant colonies can accomplish complex tasks that in some cases far exceed the individual capabilities of a single ant.

The field of Ant algorithms studies models derived from the observation of real ants' behavior and uses these models as a source of inspiration for the design of novel algorithms for the solution of optimization and distributed control problems. The main idea is that the self-organizing principles which allow the highly coordinated behavior of real ants can be exploited to coordinate populations of artificial agents that collaborate to solve computational problems. Several different aspects of the behavior of ant colonies have inspired different kinds of ant algorithms. Examples are foraging, division of labor, brood sorting, and cooperative transport. In all these examples, ants coordinate their activities via stigmergy, a form of indirect communication mediated by modifications of the environment. For example, a foraging ant deposits a chemical on the ground which increases the probability that other ants will follow the same path.

Biologists have shown that many colony-level behaviors observed in social insects can be explained via rather simple models in which only stigmergic communication is present. In other words. Biologists have shown that it is often sufficient to consider stigmergic, indirect communication to explain how social insects can achieve self- organization. The idea behind ant algorithms is then to use a form of artificial to coordinate societies of artificial agents.

One of the most successful examples of ant algorithms is known as "Ant Colony Optimization" or ACO. ACo is inspired by the foraging behavior of ant colonies, and targets discrete optimization problems.

The visual perceptive faculty of many ant species is only rudimentarily developed and there are ant species that are completely blind. In fact, an important insight of early research on ants' behavior was that most of the communication among individuals, or between individuals and the environment is based on the use of chemicals produced by the ants. These chemicals are called pheromones. This is different from, for example, what happens in humans and in other higher species, whose most important senses are visual or acoustic. Particularly important for the social life of some ant species is the trail pheromone. Trail pheromone is a specific type of pheromone that some ant species, such as Lasius Niger or the Argentine ant Iridomyrmex humilis use for marking paths on the ground, for example, paths from food sources to the nest. By sensing pheromone trails forgers can follow the path to food discovered by other ants. This collective trail-laying and trail-following behavior whereby and ant is influenced by a chemical trail left by other ants is the inspiring source of ACO.

Ant Colony Optimization (ACO) is a technique of Swarm Intelligence that studies artificial systems that take inspiration from the behavior of real ant colonies and which are used to solve discrete optimization problems.

The complex social behaviors of ants have been much studied by Science, and Computer scientists are now finding that these behavior patterns can provide models for solving difficult combinatorial optimization problems. The attempt to develop algorithms inspired by one aspect of ant behavior, the ability to find what computer scientists would call shortest paths, has become the field of ant colony optimization (ACO), the most successful and widely recognized algorithmic technique based on ant behavior.

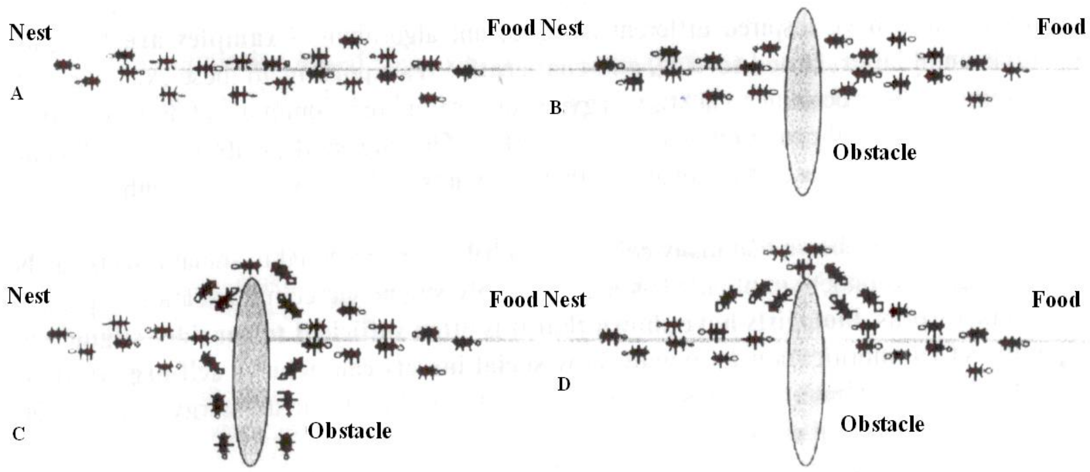


Fig 1. A Real ants follow a path between Nest and Food source B, an obstacle appears on the path.

ACO is a Meta heuristic technique that is quite successful in solving many combinatorial optimization problems. In ACO artificial ants build solutions by moving on the problem graph and they, mimicking real ants, deposit artificial pheromone on the graph in such a way that future artificial ants can build better solutions. ACO has been successfully applied to and Impressive number of optimization problems.

DOI 10.17148/IARJSET.2020.7101

The inspiring source of ACO is the foraging behavior of real ants which enables them to find shortest paths between food sources and their nests. Ants tend to deposit a substance called pheromone while walking from their nests to food source and vice versa. Paths that were marked by stronger pheromone concentrations are chosen with higher probability than that passes weaker pheromone.

2.2.1 CHARACTERISTICS OF ACO

This new heuristic has the following desirable characteristics-

1. It. Is versatile, in that it can be applied to similar versions of the same problem: for example, there is a straightforward extension from the traveling salesman problem (TSP) to the Asymmetric Traveling Salesman Problem (ATSP).
2. It is robust. It can be applied with minimal changes to other combinatorial optimization problems such as the quadratic assignment problem (QAP) and the job Scheduling Problem (JSP).
3. It is a population based approach. It uses several agents simultaneously to arrive at a solution. This allows the use of positive feedback as a such mechanism. It also makes the system a candidate for parallel implementations.

Despite these positive properties, for some applications, the Ant System can be outperformed by more specialized algorithms. Still, it is useful for problems which have peculiarities which make the application of the standard best performing algorithm impossible.

In this algorithm, the search activities are distributed over so called “ants”, that is, agents with very simple basic capabilities which, to some extent mimic the behavior of real ants. An isolated ant moves randomly, an ant encountering a previous laid trail can detect it and decide with high probability to follow it, thus reinforcing the trail with its own pheromone. The more the ants following a trail, the more attractive that trail becomes for being followed. The process is thus characterized by a positive feedback loop, where the probability with which an ant chooses a path increases with the number of ants that previously chose the same path.

2.2.2 APPLICATIONS OF ACO:

- To find shortest paths.
- To find approximate solutions to difficult combinatorial optimization problems.
- Ant net and ACO algorithm specially designed for the network routing problem, i.e. building & maintaining routing tables in a packet switched telecommunication network.
- Discovery of classification rules using ACO
- Web session clustering using ACO.
- Rerouting traffic in a busy communication network.

III. THE ORIGINS OF ANT COLONY OPTIMIZATION

Marco Dorigo and colleagues introduced the first ACO algorithms in the early 1990’s [30, 34, 35]. The development of these algorithms was inspired by the observation of ant colonies. Ants are social insects. They live in colonies and their behavior is governed by the goal of colony survival rather than being focused on the survival of individuals. The behavior that provided the inspiration for ACO is the ants’ foraging behavior, and in particular, how ants can find shortest paths between food sources and their nest. When searching for food, ants initially explore the area surrounding their nest in a random manner. While moving, ants leave a chemical pheromone trail on the ground. Ants can smell pheromone. When choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the quantity of pheromone that an ant leaves on the ground may depend on the quantity and quality of the food. The pheromone trails will guide other ants to the food source. It has been shown in [27] that the indirect communication between the ants via pheromone trails—known as *stigmergy* [49]—enables them to find shortest paths between their nest and food sources. This is explained in an idealized setting in Fig. 1.

As a first step towards an algorithm for discrete optimization we present in the following a discretized and simplified model of the phenomenon explained in Fig. 1. After presenting the model we will outline the differences between the model and the behavior of real ants. Our model consists of a graph $G = (V, E)$, where V consists of two nodes, namely v_s (representing the nest of the ants), and v_d (representing the food source). Furthermore, E consists of two links, namely e_1 and e_2, between v_s and v_d. To e_1 we assign a length of l_1, and to e_2 a length of l_2 such that $l_2 > l_1$. In other words, e_1 represents the short path between v_s and v_d, and e_2 represents the long path. Real ants deposit pheromone on the paths on which they move. Thus, the chemical pheromone trails are modeled as follows. We introduce an artificial pheromone value τ_i for each of the two links e_i, $i = 1, 2$. Such a value indicates the strength of the pheromone trail on the corresponding path. Finally, we introduce n_a artificial ants. Each ant behaves as follows: Starting from v_s (i.e., the nest),

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

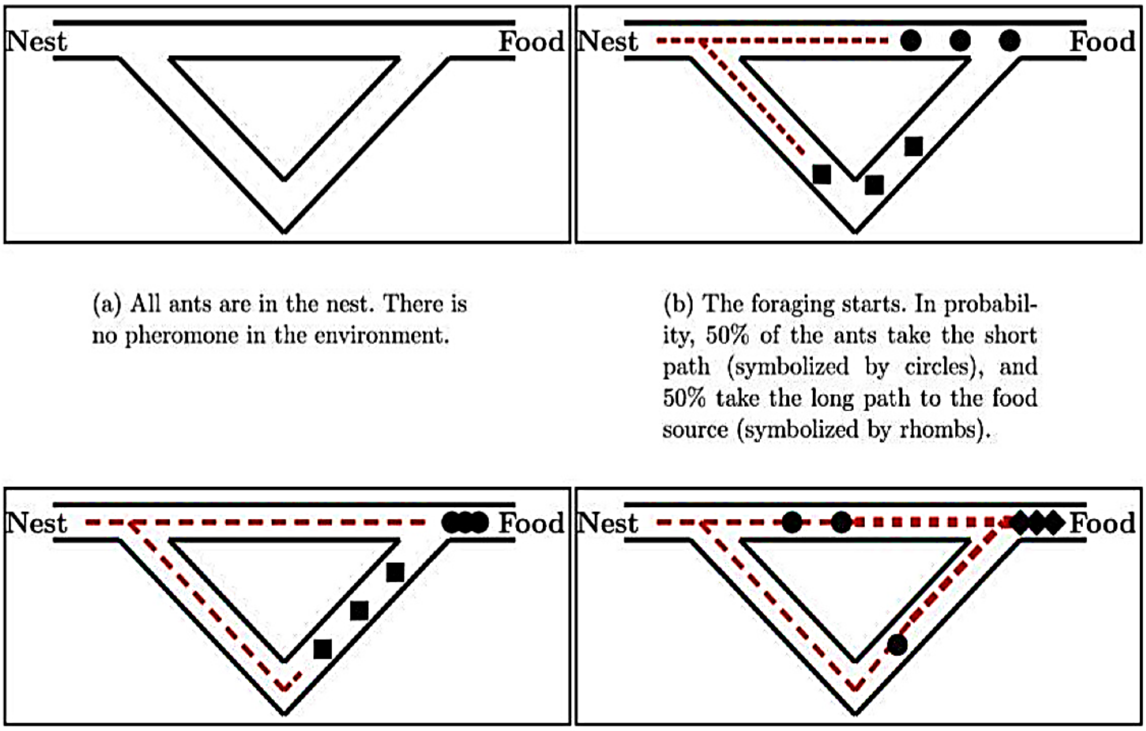
International Advanced Research Journal in Science, Engineering and Technology

Vol. 7, Issue 1, January 2020

an ant chooses with probability between path e_1 and path e_2 for reaching the food source v_d. Obviously, if $\tau_1 > \tau_2$, the probability of choosing e_1 is higher, and vice versa. For returning from v_d to v_s, an ant uses the same path as it chose to reach v_d,[4] and it changes the artificial pheromone value associated to the used edge.

$$p_i = \frac{\tau_i}{\tau_1 + \tau_2}, \quad i = 1, 2, \tag{1}$$

More in detail, having chosen edge e_i an ant changes the[4] Note that this can be enforced because the setting is symmetric, i.e., the choice of a path for moving from v_s to v_d is equivalent to the choice of a path for moving from v_d to v_s.



(a) All ants are in the nest. There is no pheromone in the environment.

(b) The foraging starts. In probability, 50% of the ants take the short path (symbolized by circles), and 50% take the long path to the food source (symbolized by rhombs).

(c) The ants that have taken the short path have arrived earlier at the food source. Therefore, when returning, the probability to take again the short path is higher.

(d) The pheromone trail on the short path receives, in probability, a stronger reinforcement, and the probability to take this path grows. Finally, due to the evaporation of the pheromone on the long path, the whole colony will, in probability, use the short path.

Fig. 1. An experimental setting that demonstrates the shortest path finding capability of ant colonies. Between the ants' nest and the only food source exist two paths of different lengths. In the four graphics, the pheromone trails are shown as dashed lines whose thickness indicates the trails' strength.

Artificial pheromone value τ_i as follows:

$$\tau_i \leftarrow \tau_i + \frac{Q}{l_i}, \tag{2}$$

Where the positive constant Q is a parameter of the model. In other words, the amount of artificial pheromone that is added depends on the length of the chosen path: the shorter the path, the higher the amount of added pheromone.
The foraging of an ant colony is in this model iteratively simulated as follows: At each step (or iteration) all the ants are initially placed in node v_s. Then, each ant moves from v_s to v_d as outlined above. As mentioned in the caption of Fig. 1(d),

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

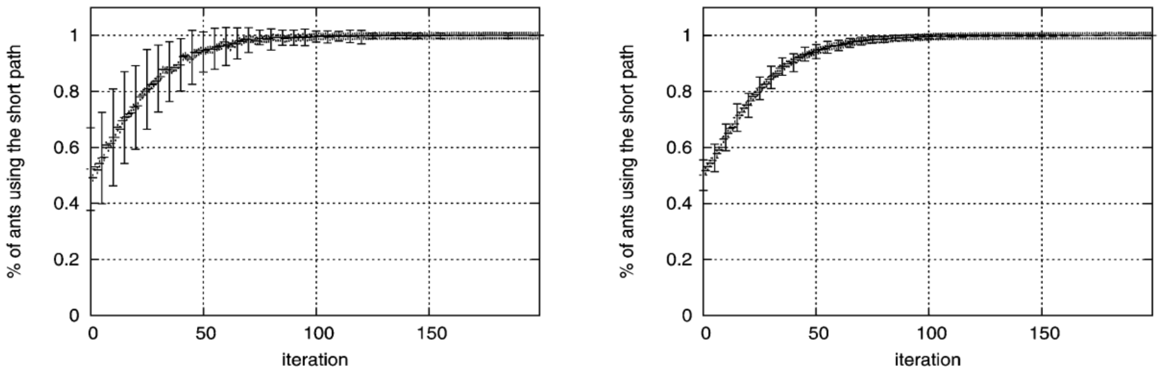
International Advanced Research Journal in Science, Engineering and Technology

Vol. 7, Issue 1, January 2020

in nature the deposited pheromone is subject to an evaporation over time. We simulate this pheromone evaporation in the artificial model as follows:

$$\tau_i \leftarrow (1-\rho)\cdot\tau_i, \quad i = 1, 2. \tag{3}$$

The parameter $\rho \in (0, 1]$ is a parameter that regulates the pheromone evaporation. Finally, all ants conduct their return trip and reinforce their chosen path as outlined above. We implemented this system and conducted simulations with the following settings: $l_1 = 1$, $l_2 = 2$, $Q = 1$. The two pheromone values were initialized to 0.5 each. Note that in our artificial system we cannot start with artificial pheromone values of 0. This would lead to a division by 0 in Eq. (1). The results of our simulations are shown in Fig. 2. They clearly show that over time the artificial colony of ants converges to the short path, i.e., after some time all ants use the short path. In the case of 10 ants (i.e., $n_a = 10$, Fig. 2(a)) the random fluctuations are bigger than in the case of 100 ants (Fig. 2(b)). This indicates that the shortest path finding capability of ant colonies results from a cooperation between the ants.



(a) Colony size: 10 ants

(b) Colony size: 100 ants

Fig. 2. Results of 100 independent runs (error bars show the standard deviation for each 5th iteration). The x-axis shows the iterations, and the y-axis the percentage of the ants using the short path.

The main differences between the behavior of the real ants and the behavior of the artificial ants in our model are as follows:

(1) While real ants move in their environment in an asynchronous way, the artificial ants are synchronized, i.e., at each iteration of the simulated system, each of the artificial ants moves from the nest to the food source and follows the same path back.

(2) While real ants leave pheromone on the ground whenever they move, artificial ants only deposit artificial pheromone on their way back to the nest.

(3) The foraging behavior of real ants is based on an implicit evaluation of a solution (i.e., a path from the nest to the food source). By implicit solution evaluation we mean the fact that shorter paths will be completed earlier than longer ones, and therefore they will receive pheromone reinforcement more quickly. In contrast, the artificial ants evaluate a solution with respect to some quality measure which is used to determine the strength of the pheromone reinforcement that the ants perform during their return trip to the nest.

IV. CONCLUSION

The general idea underlying the Ant System paradigm is that of a population of agents, each guided by and autocatalytic process directed by a greedy force. Were an agent alone, the autocatalytic process and the greedy force would tend to make the agent converge to a suboptimal tour with exponential aped. When agents interact, it appears that the greedy force can give the right suggestions to the autocatalytic process and facilitate quick convergence to very good, often optimal, solutions without getting stuck in local optima.

REFERENCES

[1]. Alupoaei S, Katkoori S. Ant colony system application to marcocell overlap removal. IEEE Trans Very Large Scale Integr (VLSI) Syst 2004;12(10):1118–22.

[2]. Bautista J, Pereira J. Ant algorithms for assembly line balancing. In: Dorigo M, Di Caro G, Sampels M, editors. Ant algorithms—Proceedings of

ANTS 2002—Third international workshop. Lecture Notes in Comput Sci, vol. 2463. Berlin: Springer; 2002. p. 65–75.
[3]. Bertsekas DP, Tsitsiklis JN, Wu C. Rollout algorithms for combinatorial optimization. J Heuristics 1997;3:245–62.
[4]. Bianchi L, Gambardella LM, Dorigo M. An ant colony optimization approach to the probabilistic traveling salesman problem. In: Merelo JJ, Adamidis P, Beyer H-G, Fernández-Villacanas J-L, Schwefel H-P, editors. Proceedings of PPSN-VII, seventh international conference on parallel problem solving from nature. Lecture Notes in Comput Sci, vol. 2439. Berlin: Springer; 2002. p. 883–92.
[5]. Bilchev B, Parmee IC. The ant colony metaphor for searching continuous design spaces. In: Fogarty TC, editor. Proceedings of the AISB workshop on evolutionary computation. Lecture Notes in Comput Sci, vol. 993. Berlin: Springer; 1995. p. 25–39.
[6]. Birattari M, Di Caro G, Dorigo M. Toward the formal foundation of ant programming. In: Dorigo M, Di Caro G, Sampels M, editor. Ant algorithms-Proceeding of ANTS 2002-Third international workshop. Lecture Notes in Comp Sci, vol. 2463. Berlin: Springer;2002. p.188-201
[7]. Blesa M, Blum C. Ant colony optimization for the maximum edge-disjoint paths problem. In: Raidl GR, Cagnoni S, Branke J, Corne DW, Drechsler R, Jin Y, Johnson CG, Machado P, Marchiori E, Rothlauf R, Smith GD, Squillero G, editors. Applications of evolutionary comput- ing, proceedings of EvoWorkshops 2004. Lecture Notes in Comput Sci, vol. 3005. Berlin: Springer; 2004. p. 160–9.
[8]. Blum C. Theoretical and practical aspects of Ant colony optimization. Dissertations in Artificial Intelligence. Berlin: Akademische Verlags-gesellschaft Aka GmbH; 2004.
[9]. Blum C. Beam-ACO—Hybridizing ant colony optimization with beam search: An application to open shop scheduling. Computers & Operations Res 2005;32(6):1565–91.
[10]. Blum C, Blesa MJ. Combining ant colony optimization with dynamic programming for solving the k-cardinality tree problem. In: Cabestany J, Prieto A, Sandoval F, editors. 8th international work-conference on artificial neural networks, computational intelligence and bioinspired systems (IWANN’05). Lecture Notes in Comput Sci, vol. 3512. Berlin: Springer; 2005. p. 25–33.
[11]. Blum C, Dorigo M. The hyper-cube framework for ant colony optimization. IEEE Trans Syst Man Cybernet Part B 2004;34(2):1161–72.
[12]. Blum C, Dorigo M. Search bias in ant colony optimization: On the role of competition-balanced systems. IEEE Trans Evolutionary Comput 2005;9(2):159–74.
[13]. Blum C, Roli A. Metaheuristics in combinatorial optimization: Overview & conceptual comparison. ACM Comput Surveys 2003;35(3):268-308
[14]. Blum C, Sampels M. An ant colony optimization algorithm for shop scheduling problems. J Math Modelling Algorithms 2004;3(3):285–308.
[15]. Blum C, Socha K. Training feed-forward neural networks with ant colony optimization: An application to pattern classification. In: Proceedings of the 5th international conference on hybrid intelligent systems (HIS). Piscataway, NJ: IEEE Press; 2005. p. 233–8.
[16]. Bonabeau E, Dorigo M, Theraulaz G. Swarm intelligence: From natural to artificial systems. New York: Oxford University Press; 1999.
[17]. Bonabeau E, Dorigo M, Theraulaz G. Inspiration for optimization from social insect behavior. Nature 2000;406:39–42.
[18]. Box GEP, Muller ME. A note on the generation of random normal deviates. Ann Math Statist 1958;29(2):610–1.
[19]. Brandt A. Multilevel computations: Review and recent developments. In: McCormick SF, editor. Multigrid methods: Theory, applications, and supercomputing, proceedings of the 3rd Copper Mountain conference on multigrid methods. Lecture Notes in Pure and Appl Math, vol. 110. New York: Marcel Dekker; 1988. p. 35–62.
[20]. Bui TN, Rizzo JR. Finding maximum cliques with distributed ants. In: Deb K, et al., editors. Proceedings of the genetic and evolutionary computation conference (GECCO 2004). Lecture Notes in Comput Sci, vol. 3102. Berlin: Springer; 2004. p. 24–35.
[21]. Bullnheimer B, Hartl R, Strauss C. A new rank-based version of the Ant System: A computational study. Central European J Operations Res Econom 1999;7(1):25–38.
[22]. Campos M, Bonabeau E, Theraulaz G, Deneubourg J-L. Dynamic scheduling and division of labor in social insects. Adapt Behavior 2000;8(3):83–96.
[23]. Corry P, Kozan E. Ant colony optimisation for machine layout problems. Comput Optim Appl 2004;28(3):287–310.
[24]. Costa D, Hertz A. Ants can color graphs. J Oper Res Soc 1997;48:295–305.
[25]. Della Croce F, Ghirardi M, Tadei R, Recovering beam search: enhancing the beam search approach for combinatorial optimisation problems. In: Proceedings of PLANSIG 2002—21th workshop of the UK planning and scheduling special interest group; 2002. p. 149–69.
[26]. Den Besten ML, Stützle T, Dorigo M. Ant colony optimization for the total weighted tardiness problem. In: Schoenauer M, Deb K, Rudolph G, Yao X, Lutton E, Merelo JJ, Schwefel H-P, editors. Proceedings of PPSN-VI, sixth international conference on parallel problem solving from nature. Lecture Notes in Comput Sci, vol. 1917. Berlin: Springer; 2000. p. 611–20.
[27]. Deneubourg J-L, Aron S, Goss S, Pasteels J-M. The self-organizing exploratory pattern of the Argentine ant. J Insect Behaviour 1990;3:159–68.
[28]. Di Caro G, Dorigo M. AntNet: Distributed stigmergetic control for communications networks. J Artificial Intelligence Res 1998;9:317–65.
[29]. Doerner K, Gutjahr WJ, Hartl RF, Strauss C, Stummer C. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolioselection. Ann Oper Res 2004;131:79–99.
[30]. Dorigo M, Optimization, learning and natural algorithms. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992 [in Ital- ian].