# Hybrid Swarm Intelligence

A

**Tad Gonsalves**
*Department of Information and Communication Sciences, Sophia University, Japan*

## INTRODUCTION

In recent years, there has been a sudden boom in the application of metaheuristic algorithms to solve optimization problems. The metaheuristic techniques differ from the mathematical programming techniques in that they do not use the gradient of the objective function but heuristic search. The heuristic and meta-heuristic search algorithms are a trial-and-error approach for solving decision-making problems. They employ a rule of thumb with the expectation of reaching the optimum solution, though there is no guarantee for it. A heuristic is a problem-dependent algorithm that exploits problem dependent information to find a sufficiently good solution to a specific problem, while a metaheuristic is a general-purpose algorithm that can be applied to almost any type of optimization problem (Saka et al., 2013; Boussaid et al., 2013).

Some of the prominent metaheuristic techniques are based on Swarm Intelligence (SI). A swarm is a large number of homogenous, unsophisticated agents that interact locally among themselves and their environment, without any central control or management to yield a global behavior. The collective behavior of decentralized and self-organized natural or artificial systems that leads to the solution of complex problems is called swarm intelligence (Kennedy et al., 2001). It is swarm intelligence that enables a colony of ants, for example, to find the shortest route between the nest and the food source or a swarm of bees to spot the locality with maximum amount of nectar.

The two most prominent SI optimization techniques are: Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). PSO, proposed by Kenedy and Eberhart (1995), and ACO proposed by Dorigo (1992) have become very popular in solving complex and intricate optimization problems in various fields (Yang et al., 2009). Recent developments in ACO and PSO algorithms and their robust industrial applications are found in Chan et al. (2007).

However, despite having several attractive features, it has been observed that these algorithms do not always perform as expected. According to the *No Free Lunch Theorem*, an algorithm which performs exceptionally well on one problem will perform poorly on another, so much so that the average performance of all metaheuristic algorithms is roughly the same (Wolpert, 1997). PSO, for instance, although performs well on most test and application problems, has the weakness of converging prematurely and falling in local optima. ACO, too, has similar limitations. Several PSO and ACO variants have been developed to overcome these limitations.

The success of the metaheuristics optimization algorithms depends to a large extent on the careful balance between two conflicting goals: exploration (diversification) and exploitation (intensification). While exploration is important to ensure that every part of the solution domain is searched enough to provide a reliable estimate of the global optimum; exploitation, on the other hand, is important to concentrate the search effort around the best solutions found so far by searching their neighborhoods to reach better solutions. The search algorithms achieve these two goals by using local or global search approaches, or an integration of both.

This article deals with the hybridization of PSO with ACO, since the two are arguably the most outstanding techniques of the SI paradigm. It identifies four different types of ACO-PSO hybrid strategies: *Parallel* hybrid (the two algorithms work independently to search for the global optimum), *global-local* hybrid (one of the algorithms does the global search and the other local search), *compound* hybrid (the two algorithms

DOI: 10.4018/978-1-4666-5888-2.ch018

themselves are hybridized with other metaheuristic algorithms) and *classical* hybrid (consisting of the classical discrete ACO and the classical continuous PSO).

This article is divided into the following sections: The section on Background describes the background of the swarm intelligence techniques. The section on Swarm Intelligence introduces the ACO and PSO algorithms together with their variants. The section on Hybrid Swarm Intelligence describes the four different ACO-PSO hybrid strategies mentioned above. The section on Applications of ACO-PSO hybrid algorithms explains the hybrid algorithm applications on benchmark problems, in engineering, business and data mining. The section on Future Research Direction indicates the areas of further development of the hybrid algorithms. The article ends with a short conclusion and a list of additional readings.

## BACKGROUND

Swarm Intelligence is a distinct research area in the newly developed field of Computational Intelligence (CI). It refers to a collection of techniques based on the collective behavior in decentralized, self-organized systems. A swarm is a loosely structured collection of interacting agents, natural or artificial. They contribute to the group and in turn benefit from the group, by communicating and interacting with each other.

The agents in the swarm interact and co-ordinate their activities through a mechanism called *stigmergy*. This means that the actions of agents leave a trace in the environment which stimulates the actions of the other agents in the swarm. In that way, subsequent actions tend to reinforce and build on each other, leading to the spontaneous emergence of coherent and self-organized activity in the swarm.

Computational models of swarms are based on the group behavior found in nature. Some of the well-known swarm behaviors found in nature are flocks of birds, school of fish, swarm of bees and colonies of ants. PSO and ACO are the two well-known and widely used computational swarm intelligence algorithms. The former mimics the nectar-collecting behavior of bees, while the latter mimics the stigmergy of an ant colony foraging for food.

## SWARM INTELLIGENCE

### Ant Colony Optimization

The Ant Colony Optimization (ACO) which is based on the foraging behavior of ants was first proposed by Dorigo (Dorigo, 1992). A generic ACO algorithm is shown in Figure 1.

In step 1, the algorithm parameters are initialized and all the artificial ants (random solutions) are generated. The loop from lines 2 through 6 is repeated until the termination condition is met. The steps inside the loop consist of evaluating the solutions, updating the pheromones and constructing new solutions from the previous solutions. The two main steps inside the loop are further described below.

#### Solution Construction

Ant *k* on node *i* selects node *j,* based on the probability, $p_{ij}$, given by:

$$p_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{j \in \mathcal{N}_i^k} [\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}} & \text{if } j \in \mathcal{N}_i^k, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $\mathcal{N}_i^k$ denotes the set of candidate sub-solutions; $\tau_{ij}$ and $\eta_{ij}$ denote, respectively, the pheromone value and the heuristic value associated with $e_{ij}$.

#### Updating the Pheromone

The pheromone update operator employed for updating the pheromone value of each edge $e_{ij}$ is defined as

*Figure 1. Generic ACO algorithm*

```
1    Initialize parameters and solutions
2    While the termination criterion is not met
3        Evaluate solutions
4        Update pheromone
5        Construct new solutions
6    End
7    Output the optimum solution
```