# The Bees Algorithm as a Biologically Inspired Optimisation Method

A

**D.T. Pham**
*School of Mechanical Engineering, University of Birmingham, UK*

**M. Castellani**
*School of Mechanical Engineering, University of Birmingham, UK*

## INTRODUCTION

This article describes the Bees Algorithm in standard formulation, and presents two applications to real-world continuous optimisation engineering problems. In the first case (Pham et al., 2008; Fahmy et al., 2012), the Bees Algorithm is employed to train three artificial neural networks (ANNs) (Pham & Liu, 1995) to model the inverse kinematics of the joints of a three-link manipulator.

## BACKGROUND

The Bees Algorithm is related to several other nature-inspired population-based optimisation procedures, such as Evolutionary Algorithms (EAs) (Fogel 2000) and Swarm Intelligence (SI) (Kennedy, 2006). Whilst EAs rely on competition amongst agents to evolve the population's fitness, SI mostly models cooperation in social insects. Moreover, standard EAs rely on centralised population selection, mating, and replacement procedures, whereas SI prescribes a fully decentralised population structure, self-organisation, and distributed intelligence.

Similarly to the SI approach, the Bees Algorithm relies on the cooperation of the artificial bees to discover the optimal solution. However, the Bees Algorithm has a centralised control structure to allocate the sampling opportunities in the search space. For this reason, the Bees Algorithm lies at the intersection between the EA and SI approaches.

## THE BEES ALGORITHM

The Bees Algorithm (Pham et al., 2006a; Pham & Castellani 2009) is a nature-inspired optimisation method based on the foraging behaviour of honey bees. It uses a population of agents (artificial bees) to explore randomly the solution space, looking for regions of high performance. These regions are selected for more detailed local search. The Bees Algorithm repeats cycles of global (random) and local search until an acceptable solution is discovered, or a given number of iterations have elapsed. In its standard formulation (Pham & Castellani 2009), the Bees Algorithm makes no assumption on the nature of the solution space, such as its derivability or continuity. For this reason it is applicable to a wide range of continuous and combinatorial problems. Henceforth, unless explicitly stated, continuous optimisation problems will be considered.

### Bees Foraging Process in Nature

In a bee colony, a portion of the population explores the environment surrounding the hive in search of food (von Frisch, 1976). These scouts look for patches of flowers where pollen or nectar is easily available and rich in sugar. Upon their return to the hive, the scouts unload the food they collected. Scouts that discovered a high-quality flower patch communicate to idle foragers the position of their find through a ritual called the *waggle dance* (Seeley, 1996). The duration of the waggle dance depends on the scout's quality rating of the patch: highly rated patches are advertised via long

DOI: 10.4018/978-1-4666-5888-2.ch027

dances, which mobilise a large number of free foragers (Seeley, 1996). Once the waggle dance is terminated, the scout returns to the food source together with the recruited foragers. When they return to the hive, recruited bees may in turn waggle dance to call more workers on the food source. Thanks to this autocatalytic mechanism, a bee colony is able to exploit efficiently the most profitable food sources (Tereshko & Lee, 2002).

## The Algorithm

Without loss of generality, let us consider a minimisation problem $f(U)\rightarrow R^{+}$ defined over the n-dimensional continuous parameter space $U=\{x\in R^{n}; max_i<x_i<min_i, i=1,\ldots,n\}$. The function $f(U)$ (*fitness function*) may feature constraints on the parameters, and defines a measure of the 'cost' of each solution $x=x1\ldots xn\in U$. The aim of the optimisation task is to find the solution $\mu\in U$ of minimal cost (i.e. maximal fitness).

$$F(\mu)\leq f(x)\ (\forall x\in U) \qquad (1)$$

The standard procedure of the Bees Algorithm was outlined by Pham and Castellani (2009), and is sketched in the flowchart of Figure 1.

At the beginning of the search, *ns* artificial scout bees are randomly initialised. Each scout lands on a candidate solution *x*, and evaluates its fitness fx. The *nb* scouts that visited the locations (solutions) of highest fitness perform the *waggle dance*. In the Bees Algorithm, the waggle dance is implemented as a deterministic recruitment procedure: the scouts that found the *ne*<*nb* top ranked (elite) solutions recruit *nre* foragers each, and the remaining *nb-ne* scouts recruit nrb≤nre foragers each. That is, more (*nre*) foragers are allocated to the exploitation of the most promising areas of the fitness landscape, and less bees (*nrb*) are assigned to the remaining sites.

Each recruited forager lands near the solution visited by the scout bee (*neighbourhood search*). The site visited by the forager is drawn with uniform probability from a neighbourhood (*flower patch*) of size $a=\{a_1,\ldots,a_n\}$, and centred on the location advertised by the scout bee. If a forager lands on a solution of higher fitness than the scout bee, the forager replaces the scout.

The remaining *ns-nb* scouts are randomly scattered on the search space (*global search*).

The algorithm repeats cycles of recruitment (waggle dance), local, and global search until a satisfactory solution is found (fitness above a threshold) or a maximum

*Figure 1. Flowchart of bees algorithm*