**Relevant background material [5]**

The Artificial Bee Colony (ABC) algorithm is an optimisation algorithm based on the foraging behaviour of the honeybee swarm. It belongs to a class of heuristics that use swarm intelligence. *Swarm intelligence* refers to the emergent collective behaviour of a swarm, a collection of (simple) individual agents.

There have been several attempts to model the behaviour of the honeybee swarm.

A bee colony system was first proposed by Tereshko and Loengarov (2005), using robot to solve tasks. They also established the first model of forage selection: interactions between found sources, employed forages, unemployed foragers, and two types of behaviour: recruitment to a nectar source and abandonment of a source.

Teodorovic (2003) later proposed a Bee Colony Optimisation Metaheuristic (BCO) capable of solving combinatorial problems.

This approach was further refined by Drias et al. (2005) into Bee Swarm Optimisation (BSO), which was adapted to solve the maximum weighted satisfiability problem.

Benatchba et al. (2005) introduced a metaheuristic to solve a 3-SAT problem based on bees’ reproduction.

Wedde et al. (2004) developed a novel routing algorithm called BeeHive.

The cited works all focus on *combinatorial* problems.

Yang (2005) developed a Virtual Bee Algorithm (VBA) to solve numeric function optimisation. A swarm of virtual bees is generated and starts somewhere random in the state space. Bees interact on finding some target nectar corresponding to the encoded values of the function; the intensity of these interactions describes good solutions.

This paper builds upon previous work by the authors in Karaboga (2005) and Basturk and Karaboga (2006) describing the ABC algorithm and comparing its performance with a GA.

**A detailed pseudocode description of the ABC algorithm [15]**

**A natural language description of the ABC algorithm [10]**

In the ABC algorithm, a colony consists of three groups of bees:

* Employee: a bee going to a food source that is previously visited. Facilitates *exploitation*, the local search for solutions in a promising area of the search space.
* Onlookers: a bee waiting in the dance area to choose a food source. Also facilitates exploitation.
* Scouts: a bee carrying out a random search for a new food source. Facilitates *exploration*, the global search for solutions in the search space.

A *food source* is defined by a *position* in the state space, a possible solution to the optimisation problem, and a *nectar amount*, the fitness of that solution. The number of food sources is termed $S\_n$. $S\_n$ is one of three control parameters in ABC. The other two are $limit$ is the number of times a food source is modified before, if no superior nearby food source is obtained, it is discarded. $C\_{max}$, the maximum cycle number, determines the number of iterations the population undergoes.

* Initialise
* REPEAT:
  + Employed bee phase
  + Onlooker bee phase
  + Scout bee phase
* UNTIL C\_max is reached

After initialisation, the population is subject to repeated cycles up to C\_max.

**Initialisation**

A randomly distributed initial population (of size $S\_n$) of food sources is generated. Each food source position is a $D$-dimensional vector, where $D$ is the number of optimisation parameters. The nectar amounts of each food source are computed.

**Employed bee phase**

Each employed bee goes the food source in her memory. She produces a probabilistic modification of her food source position and evaluates the nectar amount. If it is greater than her original food source, it is kept; otherwise, it is discarded. If a superior source is not found, the number of times this source has been searched is incremented. She then returns to the hive and shares the nectar amount of her food source to the onlooker bees.

**Onlooker bee phase**

Each onlooker bee observes the dance of each employed bee. The onlooker chooses a food source using roulette-wheel selection. She goes to that source and, following the same procedure as the employed bee, produces a modification of its position and keeps it if it is superior. If a superior source is not found, the number of times this source has been searched is incremented.

**Scout bee phase**

The food source that has been sampled the most is examined. If it has been sampled more than $limit$, a scout bee goes to a new food source.

**Modifying a food source**

v\_ij = x\_ij + phi\_ij(x\_ij + x\_kj) where k is in [1, SN}, j is in [1, D]. k is determined randomly but must be different from i. phi\_i,j is a random number between [-1, 1]. It controls the production of a neighbouring food source around x I,j and the modification represents the comparison of the neighbour food positions visually. As the difference between the parameters of x\_i, j and x\_k, j decreases, so too does the perturbation of x\_i,j. As the search approaches the optimum solution in the search space, the step length is adaptively reduced.

**Details of experiments [15]**

**Define all concepts**

A function is *multimodal* if it has two or more local optima. An optimisation algorithm must avoid over-exploiting the regions around local minima. It must balance *exploration* and *exploitation*.

**Give a very brief overview of the benchmark functions**

Five classical benchmark algorithms (Srinivasan and Seow, 2003) were used to test the performance of ABC against PSO, PS-EA, and GA:

* Griewank. The Griewank function has a product term that introduces interdependence among the variables, penalising techniques which optimise each variable independently.
* Rastrigin function. Value is 0 at its global minimum at the original. Produces many local, regularly distributed minima, so an optimisation algorithm can easily be trapped. Non-convex. The typical example of a non-linear multi-modal function.
* Rosenbrock function. Global optimum is in a long, narrow, parabolic, flat valley. It is easy to find a local optimum, but difficult to converge. Variables are therefore strongly dependent. Gradients generally do not point towards the optimum.
* Ackley function. Tests how efficiently an algorithm both explores and exploits. Has an exponential term that covers its surface with numerous local optima.
* Schwefel function. Surface comprises a great number of peaks and valleys, with the second-best minimum far from the global minimum, which itself is near the bounds of the domain.

**Give a very brief overview of the comparator algorithms.**

Particle Swarm Optimisation (PSO) is a popular population-based stochastic optimisation technique. Also in this class are Evolutionary Algorithms (EA), which simulate evolution. Genetic Algorithms (GA) are the most popular variety. The Particle Swarm Evolutionary Algorithm (PS-EA) is a hybrid of PSO and EA.

Why is this fifteen mark? I need a lot more depth, methinks.

Common control parameters of the algorithms are population size ($S\_n$ in ABC) and the number of generations ($C\_{max}$). In the experiments, a ($C\_{max}$) of 500, 750, and 1000 was used with the dimensions 10, 20, and 30, respectively. Population size was held constant at 125.

The GA used single-point uniform crossover (swapping one gene at the same point in two individuals) with a probability $P\_c = 0.95$. Individuals were selected randomly and ranked with a linear fitness function. Gaussian mutation occurred with probability $P\_m=0.1$

PSO settings are more complex. The learning factors were set to be two.

ABC settings. 50% of the colony was scouts, 50% were onlookers, and there was 1 scout. Each experiment was repeated 30 times with different random seeds, and the mean function values of the best solutions found by the algorithms are recorded.

**An overview of results [5]**

The paper found that the ABC algorithm performs better than PS-EA on the Greiwank and Ackley functions. It is only outperformed on the Schwefel function by PS-EA and GA for dimensions 20 and 30. After increasing the maximum number of cycles (MCN), the ABC algorithm converged to the minimum of the chwefel function as well.

It is successful in optimising multivariable and multimodal functions.

# <https://en.wikipedia.org/wiki/Artificial_bee_colony_algorithm>

Srinivasan, D., Seow, T.H.: Evolutionary Computation, CEC ’03, 8–12 Dec. 2003, 4(2003), Canberra, Australia, pp. 2292–2297.