# Intuition behind our improvements

Every swarm algorithm there is usually includes action of agents which can be classified as either exploitation or exploration.

Exploitation means an agent uses its current result and tries to improve it, i.e. the agent continues to go into the direction he previously went in the search tree/space.

For AFB this is can be achieved using the walk move, so a local search.

Exploration means an agent tries to find a better solution in the search space, that is not necessarily dependent on its current solution, as is the case with exploitation. Therefore, it can be seen as a global search.

For AFB, this can be achieved using the fly move.

In order for a swarm algorithm to deliver a solution which is as close as possible to the global minimum, it needs a good balance between exploitation and exploration: It needs to be able to improve a good solution and search for different solutions if the current one does not seem promising.

If exploitation is too dominant, then the algorithm might get stuck in a local minimum, as other solutions (for TSP other, vastly different tours) will not be explored frequent enough, and vice versa.

Because in AFB the balance is mostly specified by the probability of moves walk and fly, and the authors already tuned the algorithm to probabilities that work well in practice, we did not focus on changing this balance.

Instead, our focus was mainly on exploitation: Improving a good solution, which is mainly done by the introduction of 3-opt and that big birds can only joins successful birds. Even though the latter cannot be seen directly as a local search, is does lead to more (big) birds trying to improve them.

We explicitly do not modify the fly-move, selecting a random tour, as this provides us with a rich selection of other possible solution, which at the same time is completely independent of the current solution (of an agent).

Therefore, we get an algorithm that has an empirically tested balance between exploitation and exploration. At the same time, it exploits good solutions in a harsh manner while also being able to switch to completely new solutions if they prove to be better.

This process will be repeated until a solution is reached that will be close to the optimum in most cases (the algorithm converges) .

A black and orange rectangular object

Description automatically generated