When Ant Colony Optimization does not need Swarm Intelligence

Maurice Clerc
maurice.clerc@writeme.com
August 2000

I use here just one paper as reference: the meta-heuristic defined in (Dorigo and Di Caro 1999), for it is the most general description of ACO I have found. So, any result concerning this meta-heuristic should be also true for more particular ant algorithms.

I show here that the meta-heuristic can be rewritten so that it clearly appears that each ant works independently. It means no swarm is needed and no "swarm intelligence" is used.

The point is that in this algorithm ants share information only via an "external" memory (pheromones on arcs) and they usually modify indepedently this memory (see below Hypothesis 1 in Rewriting 1).

A possibility to use something which might be called "swarm intelligence" could be to have a *deposit_pheromone_on_the_visited_arc*() process which takes into account the amount of pheromone already on the arc, on a non linear way.

But a better one could be to have ants acting like real ones, which share their memories thanks to their feelers and not only thanks to pheromones. For example an ant could say to another one what better path it has found. Note that in this case we probably would have something very similar to (discrete) Particle Swarm Optimization, but it might be worthwhile to try.

ORIGINAL META-HEURISTIC

```
procedure ACO_meta-heuristic()
    while (termination_criterion_is_not_satisfied)
        schedule_activities
            ants_generation_and_activity();
            pheromone_evaporation() ;
            deamon_action(); {optional}
        end schedule_activities
    end while
end procedure

procedure ants_generation_and_activity()
    while (available_resources)
        schedule_the_creation_of_a_new_ant();
        new_active_ant();
    end while
end procedure

procedure new_active_ant() {ant lifecycle)
    initialize_ant() ;
    M = update_ant_memory();
    while (current_state # target_state)
        A = read_local_ant_routing_table();
        P = compute_transition_probabilities(A, M,Ω);
        next_state  = apply_ant_decision_policy(P, Ω);
        move_to_next_state(next_state) ;
        if(online_step-by-step.pheromone.update)
            deposit_pheromone_on_the_visited_arc();
            update_ant_routing_table();
        end if
        M =update_internal_state() ;
    end while
    if(online_delayed.pheromone.update)
        foreach visited_arc ∈ ψ do
            deposit_pheromone_on_the_visited_arc();
            update_ant_routing_table();
        end foreach
    end if
    die();
end procedure
```

REWRITING 1 (All loops in just one procedure)

```
procedure ACO_meta-heuristic()
    while (termination_criterion_is_not_satisfied)
        schedule_activities
            while (available_resources)
                schedule_the_creation_of_a_new_ant();
                initialize_ant() ;
                M = update_ant_memory();
                while (current_state # target_state)
                    A = read_local_ant_routing_table();
                    P = compute_transition_probabilities(A, M,Ω);
                    next_state = apply_ant_decision_policy(P, Ω);
                    move_to_next_state(next_state);
                    if(online_step-by-step.pheromone.update)
                        deposit_pheromone_on_the_visited_arc();
                        update_ant_routing_table();
                    end if
                    M =update_internal_state() ;
                end while
                if(online_delayed.pheromone.update)
                    foreach visited_arc ∈ ψ do
                        deposit_pheromone_on_the_visited_arc();
                        update_ant_routing_table();
                    end foreach
                end if
                die();
            end while
            pheromone_evaporation();
        end schedule_activities
    end while
end procedure
```

REWRITING 2

Hypothesis 1
deposit_pheromone_on_the_visited_arc() is not depending on the amount of pheromone already present. This is indeed the case in all applications described in the referenced paper.

Hypothesis 2
pheromone_evaporation() is linear (just multiplying the amount of pheromone by a constant ρ smaller than 1. Same remark : it is also the case in the referenced paper.

Then
a) we can define a new procedure *deposit_evaporate_pheromone_on_the_visited_arc*() so that if
x = pheromone deposited by *deposit_pheromone_on_the_visited_arc*()
x' = pheromone deposited by *deposit_evaporate_pheromone_on_the_visited_arc*()
then $x' = \rho x$

b) we can delete the procedure *pheromone_evaporation*(), and replace it, inside the **while** (available_resources) loop, by using *deposit_evaporate_pheromone_on_the_visited_arc*(), and the result is exactly the same.

```
procedure ACO_meta-heuristic()
    while (termination_criterion_is_not_satisfied)
        schedule_activities
            while (available_resources)
                schedule_the_creation_of_a_new_ant();
                initialize_ant() ;
                M = update_ant_memory();
                while (current_state # target_state)
                    A = read_local_ant_routing_table();
                    P = compute_transition_probabilities(A, M,Ω);
                    next_state = apply_ant_decision_policy(P, Ω);
                    move_to_next_state(next_state);
                    if(online_step-by-step.pheromone.update)
                        deposit_evaporate_pheromone_on_the_visited_arc()
                    ;
                        update_ant_routing_table();
                    end if
                    M =update_internal_state() ;
                end while
                if(online_delayed.pheromone.update)
                    foreach visited_arc ∈ ψ do
                        deposit_evaporate_pheromone_on_the_visited_arc()
                    ;
                        update_ant_routing_table();
                    end foreach
                end if
                die();
            end while
        end schedule_activities
    end while
end procedure
```

REWRITING 3
To summarize, we have

```
procedure ACO_meta-heuristic()
        while (termination_criterion_is_not_satisfied)
                schedule_activities
                        while (available_resources)
                                <new_ant>
                                <ant does something>
                                <ant dies>
                        end while
                end schedule_activities
        end while
end procedure
```

Conclusion

There is no swarm at all in this procedure and, therefore, no « swarm intelligence ».

Reference

Dorigo, M. and G. Di Caro (1999). Ant Colony Optimisation: A New Meta-Heuristic. Congress on Evolutionary Computation, Washington D.C.