Report for the project for the Swarm Intelligence course

Student **Aldar Saranov (000435170 ULB)**
Aldar.Saranov@ulb.ac.be

Wednesday 30$^{\text{th}}$ May, 2018

1 Implementation

In the literature many different methods are proposed and researched including various implementations of the simulated annealing, taboo search, hybrid genetic-taboo search. However, for this project an algorithm known as Hybrid Ant System for the Quadratic Assignment Problem (HAS-QAP) was used as it was proposed by Gambardella and Dorigo in [?]. As all the ACO algorithms it uses the notion of solution construction biasing by means of pheromone trails, deposited by ants. The high-level outline of HAS-QAP is shown in Figure 1. It was implemented in two versions - with Rank-based Ant System and Elitist Ant System as different pheromone trail update techniques.

Let n be the number of facilities/locations, i.e. the size of a problem. Every solution of a QAP problem is a permutation ψ of an integer sequence from 1 to n.

The HAS-QAP includes such components as:

- Random solution generation
- Local Search
- Pheromone trail swaps
- Intensification
- Pheromone update
- Diversification

Listing 1: HAS-QAP pseudo-code

```
1  procedure HAS–QAP
2  generate m random permutations $\psi^1$ , ... , $\psi^m$ .
3  [optional] improve $\psi^1$ , ... , $\psi^m$ by local search
4  let $pi^*$ be the best solution
5  initialize the pheromone trail matrix T
6  activate intensification
7
8  while (there is time left)
9    for k from 1 to m
10     $\hat{\psi}^k$ = PheromoneTrailSwaps($\psi^k$)
11     [optional] improve $\hat{\psi}^k$ by local search to get $\tilde{\psi}^k$
12   end
13
14   for k from 1 to m
15     if intensification is active then
16       $\psi^k$ = best($\psi^k$;$\tilde{\psi}^k$)
17       if none of $\psi^k$ changed then
18         disable intensification
19     else
20       $\psi^k$ = $\tilde{\psi}^k$
21   end
22
23   if exists $\tilde{\psi}^k$ better then $\psi^*$
24     update the new best $\psi^*$ = $\tilde{\psi}^k$
25     activate intensification
26   end
27
28   update the pheromone trail matrix
29
30   if S iterations in a row are not improving then
31     perform diversification
32 end
```

Some micro-optimization were applied to the original version of HAS-QAP such as reorganizing conditional branches. For example, we extracted the conditional block on the line 15 outside the loop to avoid redundant condition checks.

1.1 Random solution

Is used in the initializing section of the algorithm. Generate m random solutions. In out implementation, the algorithm takes the facilities one by one and assigns it to one of the free locations according to random uniform rule. This is an exploration step.

1.2 Local Search

The implemented local search is based on sequential random check of all pairs i and j and performing swaps of location between the i-th and j-th facilities, in case if these swaps are profitable. For this we compute the difference of objective values before the swap and after Δ. Instead of full objective value recomputation in $O(n^2)$, one can compute Δ value in an optimized $O(n)$ way.

$$\Delta(\psi, i, j) = (b_{ij} - b_{ji}) \times (a_{\pi_i \pi_j} - a_{\pi_j \pi_i}) + \sum_{k=1}^{n} [b_{ik} \times (a_{\pi_i \pi_k} - a_{\pi_j \pi_k})$$
$$+ b_{ki} \times (a_{\pi_k \pi_i} - a_{\pi_k \pi_j}) + b_{jk} \times (a_{\pi_j \pi_k} - a_{\pi_i \pi_k}) + b_{kj} \times (a_{\pi_k \pi_j} - a_{\pi_k \pi_i})]$$

Listing 2: Local Search pseudo-code

```
procedure LocalSearch(solution ψ)
I = ∅
while (|I| < n)
  pick i uniformly randomly, i ∉ I
  J = {i}
  while (|J| < n)
    pick j uniformly randomly, j ∉ J
    if (Δ(ψ,i,j) < 0)
      exchange ψ_i and ψ_j
      J = J ∪ {j}
    end
    I = I ∪ {i}
  end
end
```