# Report for the project for the Swarm Intelligence course

Student **Aldar Saranov (000435170 ULB)**

Aldar.Saranov@ulb.ac.be

Thursday 31$^{\text{st}}$ 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 (RAS) and Elitist Ant System (EAS) 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 $\psi$ 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 ψ¹ ,... ,ψᵐ .
3  [ optional ] improve ψ¹ ,... ,ψᵐ 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        ψ̂ᵏ = PheromoneTrailSwaps(ψᵏ)
11        [ optional ] improve ψ̂ᵏ by local search to get ψ̃ᵏ
12     end
13
14     for k from 1 to m
15        if intensification is active then
16           ψᵏ = best(ψᵏ;ψ̃ᵏ)
17           if none of ψᵏ changed then
18              disable intensification
19        else
20           ψᵏ = ψ̃ᵏ
21     end
22
23     if exists ψ̃ᵏ better then ψ*
24        update the new best ψ* = ψ̃ᵏ
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

3

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

We implemented the same local search that is described in the paper. 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 $\Delta$. Instead of full objective value recomputation in $O(n^2)$, one can compute $\Delta$ value in an optimized $O(n)$ way as in Formula 1.

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

Listing 2: Local Search pseudo-code

```
1  procedure LocalSearch(solution ψ)
2  I = ∅
3  while (|I| < n)
4     pick i uniformly randomly, i ∉ I
5     J = {i}
6     while (|J| < n)
7        pick j uniformly randomly, j ∉ J
8        if (Δ(ψ, i, j) < 0)
9           exchange ψ_i and ψ_j
10          J = J ∪ {j}
11       end
12       I = I ∪ {i}
13    end
14 end
```

Thus, this local search allows only improving moves and leads to high intensification. The total local search complexity is $O(n^3)$.

## 1.3   Pheromone trail swaps

Pheromone trail value $\tau_{ij}$ is assigned to every pair of facility $i$ and location $j$. The more this value is, the more strongly the algorithm will try to bias to assigning the facility $i$ to the location $j$.

Pheromone trail swaps are applied on each iteration for each solution. These swaps have two policies - exploring and exploiting. For a given solution, an exploiting policy is applied with probability $q$. This parameter is the key parameter that defines the trade-off between exploiting and exploring.

In exploiting policy, a random facility $r$ is chosen. Then a facility $s$ is chosen, in such way, that the value $\tau_{r\pi_s}^k + \tau_{s\pi_r}^k$ is maximized. This procedure is repeated $n$ times.

In exploring policy, once again facility $r$ is chosen randomly uniformly. The facility $s$ is chosen according to a stochastic rule where the probability of choosing a facility is determined by Formula 2.

$$P(s) = \frac{\tau_{r\pi_s}^k + \tau_{s\pi_r}^k}{\sum_{j \neq r}(\tau_{r\pi_j}^k + \tau_{j\pi_r}^k)} \tag{2}$$

## 1.4 Intensification

This tool defines whether only strictly improving solutions will remain or one injects some exploration by allowing solutions, that are not the best. Initially it is activated. It is deactivated if the solutions did not a single solution changed during the current iterations, which implies that the state of stagnation was achieved. It is activated if a new best solutions was found. It corresponds to escaping a search space peak.

## 1.5 Pheromone update

As it was said RAS and EAS were implemented. In both of them the new pheromone trail values are defined as in Formula 3.

$$\tau(i+1) = (1 - \rho) \times \tau(i) + \Delta\tau(i) \tag{3}$$

In RAS the update is done according to Formulas 4, 5, 6. The idea of the RAS is to deposit pheromones according to their rank in the sorted set of all solutions and also to the best solution.

$$\Delta\tau_{ij} = \sum_{r=1}^{w-1} (w - r) \times \Delta\tau_{ij}^{r} + w \times \Delta\tau_{ij}^{bs} \tag{4}$$

$$\Delta\tau_{ij}^{r} = \begin{cases} \frac{1}{L^{r}} \text{ if arc(i,j)} \in S \\ 0 \text{ otherwise} \end{cases} \tag{5}$$

$$\Delta\tau_{ij}^{bs} = \begin{cases} \frac{1}{L^{bs}} \text{ if arc(i,j)} \in S^{bs} \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

, where $w$ - is the number of depositing ants. $S$ - current solution. $S^{bs}$ - best solution found so far.

In EAS the update is done according to Formulas 7, 8. The aim of this pheromone update is to deposit much larger amount pheromones to the best solution.

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} + e \times \Delta\tau_{ij}^{bs} \tag{7}$$

$$\Delta\tau_{ij}^{bs} = \begin{cases} \frac{1}{L^{bs}} \text{ if arc(i,j)} \in S^{bs} \\ 0 \text{ otherwise} \end{cases} \tag{8}$$

## 1.6   Diversification

Diversification is the same reset of pheromone trail values as in the initialization. The only difference is that the best solution quality found must be corrected. The pheromone trail values that is set is computed by Formula 9.

$$\tau_0 = \frac{1}{f(S^{bs})} \tag{9}$$