Ant Colony Optimization project report

Swarm Intelligence INFO-H-414

Aldar Saranov

Université libre de Bruxelles

```
1  procedure HAS-QAP
2  generate m random permutations ψ^1 ,... ,ψ^m .
3  [optional] improve ψ^1 ,... ,ψ^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     ψ̂^k = PheromoneTrailSwaps(ψ^k)
11     [optional] improve ψ̂^k by local search to get ψ̃^k
12   end
13
14   for k from 1 to m
15     if intensification is active then
16       ψ^k = best(ψ^k;ψ̃^k)
17       if none of ψ^k changed then
18         disable intensification
19     else
20       ψ^k = ψ̃^k
21   end
22
23   if exists ψ̃^k better then ψ*
24     update the new best ψ* = ψ̃^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
```

Initial solution generation

Local search

Solution modification

Applying intensification if necessary

Pheromone update (EAS and RAS)

Diversification

Initial Solution Generation

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.

Local Search

- May be applied to the initial solutions before setting the pheromone matrix and in each iteration.
- Delta value of objective values are used. They can be compute in O(n).

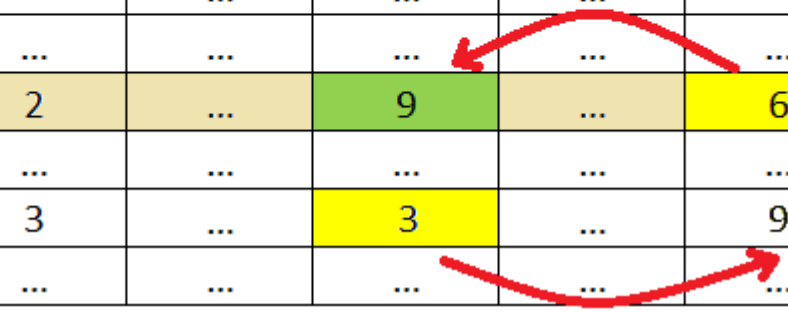
$$\Delta(\psi,i,j) = (b_{ij} - b_{ji}) \times (a_{\pi_i\pi_j} - a_{\pi_j\pi_i}) + \sum_{k=1}^{n} [b_{ik} \times (a_{\pi_i\pi_k} - a_{\pi_j\pi_k}) + b_{ki} \times (a_{\pi_k\pi_i} - a_{\pi_k\pi_j}) + b_{jk} \times (a_{\pi_j\pi_k} - a_{\pi_i\pi_k}) + b_{kj} \times (a_{\pi_k\pi_j} - a_{\pi_k\pi_i})]$$

Listing 2: Local Search pseudo-code

```
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
```

Pheromone Trail Swaps

Location / Facility	1	...	pi_s	...	pi_r	...
1	...	...	...	...	...	...
...	...	...	...	...	...	...
r	2	...	9	...	6	...
...	...	...	...	...	...	...
s	3	...	3	...	9	...
...	...	...	...	...	...	...



Exploiting policy with probability Q

$$\tau^k_{r\pi_s} + \tau^k_{s\pi_r}$$

Exploring policy with probability 1 - Q

$$P(s) = \frac{\tau^k_{r\pi_s} + \tau^k_{s\pi_r}}{\sum_{j \neq r} (\tau^k_{r\pi_j} + \tau^k_{j\pi_r})}$$

Intensification

```
for k from 1 to m
  if intensification is active then
    ψ^k = best(ψ^k ; ψ̃^k)
    if none of ψ^k changed then
      disable intensification
  else
    ψ^k = ψ̃^k
end
```

Pheromone update rule

$$\tau(i+1) = \rho \times \tau(i) + \Delta\tau(i)$$

Rank-based Ant System

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

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

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

Elitist Ant System

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

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

Diversification

Diversification is applied if during S iterations in a row there were no improvement.
Pheromone trail values are reset.
It promotes exploration.

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

Automatic tuning

The program was tuned in irace. The overall parameters description is as follows.

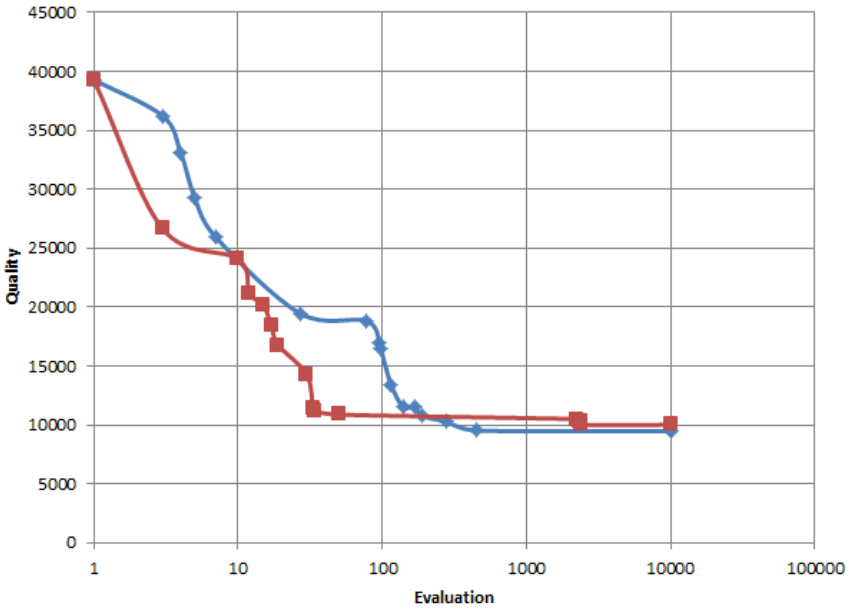
Name	Values	Description
localSearch	local-search-idsia, local-search-nonet	Whether local search is used or omitted
m	1-100	Number of ants
ρ	0.0-1.0	Weakness of evaporation
roundsReinitialize	100-10000	Numbers of non-improving rounds in a row to diversify
q	0.0-1.0	Probability of using exploiting policy
k	0.2-3.0	Selection power
factorQ	0.2-5.0	Deposit factor
σ	1-100	Number of depositing for Elitist Ant System
w	1-20	Number of depositing for Rank-based Ant System

Experimental - performance

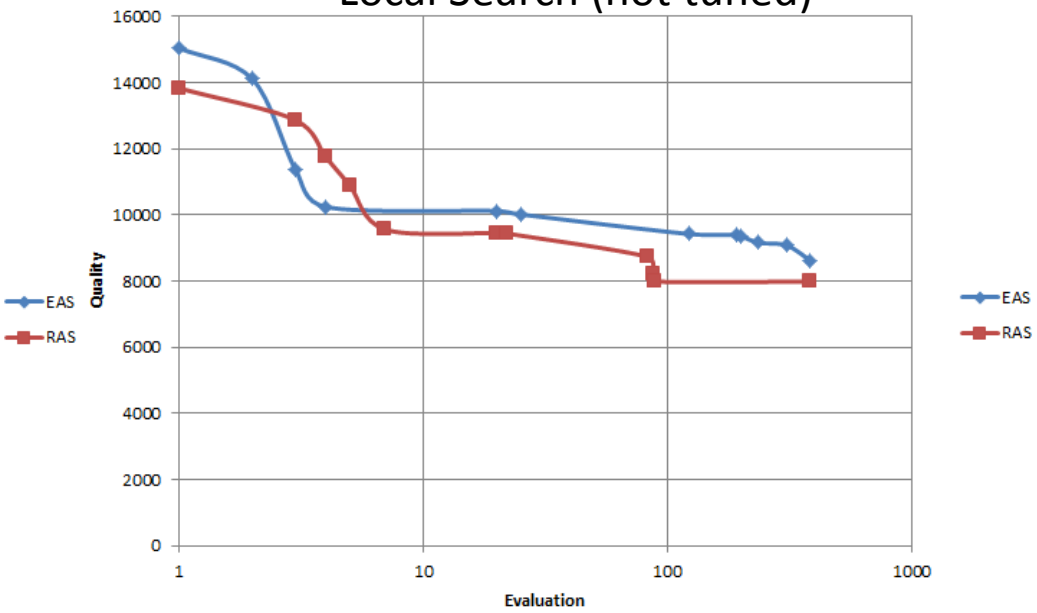
- **Not tuned.** EAS showed slightly better results with average relative solution quality 3.75%, however, it also had larger variation coefficient (meaning that some concrete instances may be solved by EAS worse than by RAS).
- **Local Search without tuning.** The configurations, however, were inherited from the previous series. RAS showed much better results with average relative solution quality 0.2% and best quality 0.17%. There is a significant improvement for both EAS and RAS if local search is added.
- **Local Search with tuning.** The results here are ambiguous, because the average relative quality of RAS is equal to 0:24% which is better than 0.66% of EAS, however, in the worst case EAS has 0:89%, whereas RAS has 1.01%.
- In all cases Wilcoxon test cannot firmly reject that the mean of the performances of these algorithms is different.

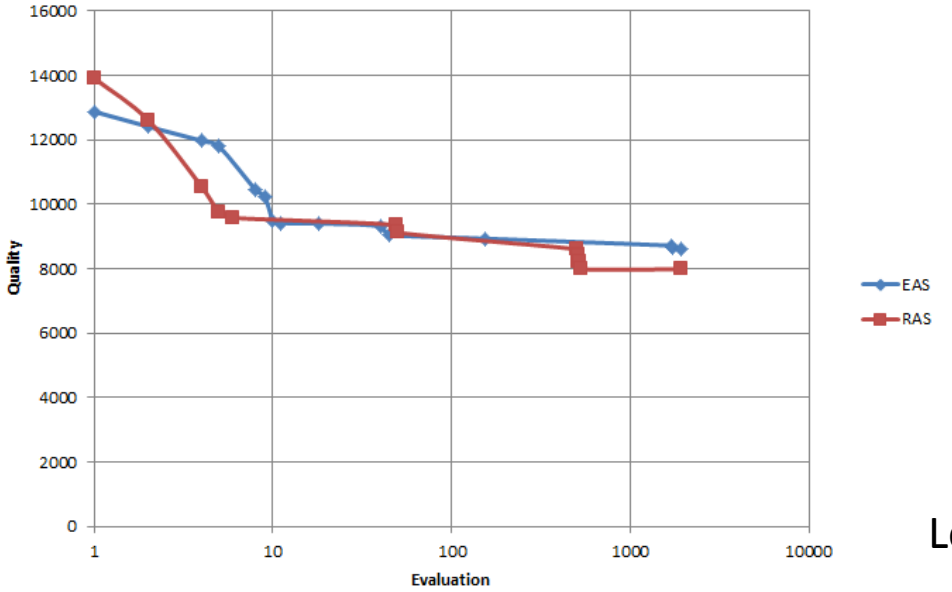
Experimental - convergence

No Local Search



Local Search (not tuned)





Local Search (tuned)