Université Libre de Bruxelles

INFO-H414

Swarm Intelligence

Ant Colonization Optimization

Daoud Yemni ULB

1er juin 2018

Table des matières

1 Algorithms

The process of of computation of the solution is simple. I use the same process used for the TSP. The representation of "tour" in TSP is an assignment for the QAP. The heuristic for the problem is defined as :

$$h_{i,j}\frac{1}{a[i,j]\cdot b[i,j]+1}$$

where a is the distance matrix and b is the flows matrix.

The computation of the probability is still the same that the basic from TSP :

$$P_{i,j}=\frac{ph[i,j]^{\alpha}\cdot h[i,j]^{\beta}}{\sum_{q\in J} ph[i,q]^{\alpha}\cdot h[i,q]^{\beta}}$$

where ph is the pheromone matrix, the denominator represents the memory of the ant.

The pheromone trails are updated as follows :

$$\begin{cases} \tau_{ij}(0)=\frac{1}{\rho\cdot N} \\ \tau_{ij}(t)\leftarrow(1-\rho)\cdot\tau_{ij}(t-1)+\Delta\tau_{ij} \end{cases}$$

where $\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}$, $\Delta\tau_{ij}^{k}=\frac{1}{L^k}$ in which L^k is the length of the solution found by the ant k and N is the number of city.

I decide to compare the Elitist Ant system and the Rank-Based Ant system. They use the same ethics that the best ant(s) can add more pheromone in the pheromone trails. Therefore, I implemented those algorithms to show which one give a better a solution for the problem of QAP.

1.1 Elitist Ant System

The Elitist Ant System is an algorithm where the goal is that only the ant(s), that have the best solution, can add a plus-value in the pheromone trails. In this algorithm, the updating function of the pheromone trails is a little bit modified. We add to the solution the number of best times the delta of the best ant(s) as follow :

$$\tau_{ij}(t)=(1-\rho)\cdot\tau_{ij}(t-1)+\sum_{k=1}^{m}\Delta\tau_{ij}^{k}+e\cdot\Delta\tau_{ij}^{bs}$$

where bs means the best solution and $\Delta\tau_{ij}^{bs} = \begin{cases} \frac{1}{L^{bs}} & if arc(i,j) \in T^{bs} \\ 0 & otherwise \end{cases}$

1.2 Rank-Based Ant System

The goal of the Rank-Based Ant algorithm is that the ants are ranked regarding their solution. Every ants can add pheromone and the impact of their pheromone depends on their rank. More the ant is ranked, more the impact of his pheromone is big.

$$\tau_{ij}(t) = (1-\rho) \cdot \tau_{ij}(t-1) + \sum_{r=1}^{\omega-1} (\omega - r)\Delta\tau_{ij}^{r} + \omega \cdot \Delta\tau_{ij}^{bs}$$

where ω is the number of top-rank ants allowed to contribute in the pheromone trails.

2 Setting parameters

For setting the parameter, The technique used in my project was inspired about the ParamILS. Given a set of configuration $\Theta = \{\theta_1, \theta_2, ..., \theta_k\}$ and a set of instances $X = \{X_1, X_2, ..., X_l\}$. I will check each configuration $\theta_i \in \Theta$ on each instance $X_j \in X$ and retrieve the best configuration for the instance θ_j. At the end, we remain the configuration that satisfied the most instances, θ_j^{best}. Then the remaining configuration θ_j^{best} is saved in a file (so one file per algorithm) and is used for the main program.

My list of predefined configuration contains all of possible combinations (except the case where $\alpha = \beta = 0$) with those values :

- Number of ants : 5, 10 or 15
- α : 0 or 1
- β : 0 or 1
- ρ : 0.5
- ω : 4

`Remark` : The technique implemented is not the best one. We can find a different best configuration each time when we apply the technique multiple times for the same algorithm. But after several runs, some of them remains often.

3 EAS vs RBA

In this section, we will compare the results obtained by the both algorithm for each instances. We run 10 times each algorithm on 10 different seeds for each instance. the results of the best solution for each seeds and each runs is stored and we compute the mean of run. The different seeds used for the experiments is :

1. $m = 5$, $\alpha = 0$ and $beta = 1$
2. $m = 5$, $\alpha = 1$ and $beta = 0$
3. $m = 5$, $\alpha = 1$ and $beta = 1$
4. $m = 10$, $\alpha = 0$ and $beta = 1$
5. $m = 10$, $\alpha = 1$ and $beta = 0$
6. $m = 10$, $\alpha = 1$ and $beta = 1$
7. $m = 15$, $\alpha = 0$ and $beta = 1$
8. $m = 15$, $\alpha = 1$ and $beta = 0$
9. $m = 15$, $\alpha = 1$ and $beta = 1$
10. $m = 20$, $\alpha = 1$ and $beta = 1$

3.1 Results

3.1.1 Best solution quality

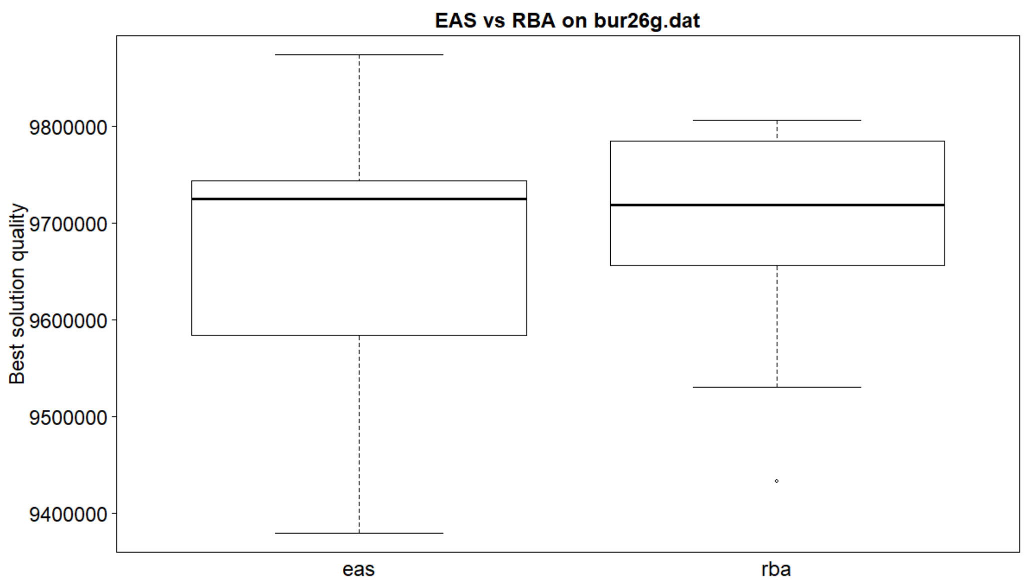

FIGURE 1 – Results on the both algorithm on the instance bur26g

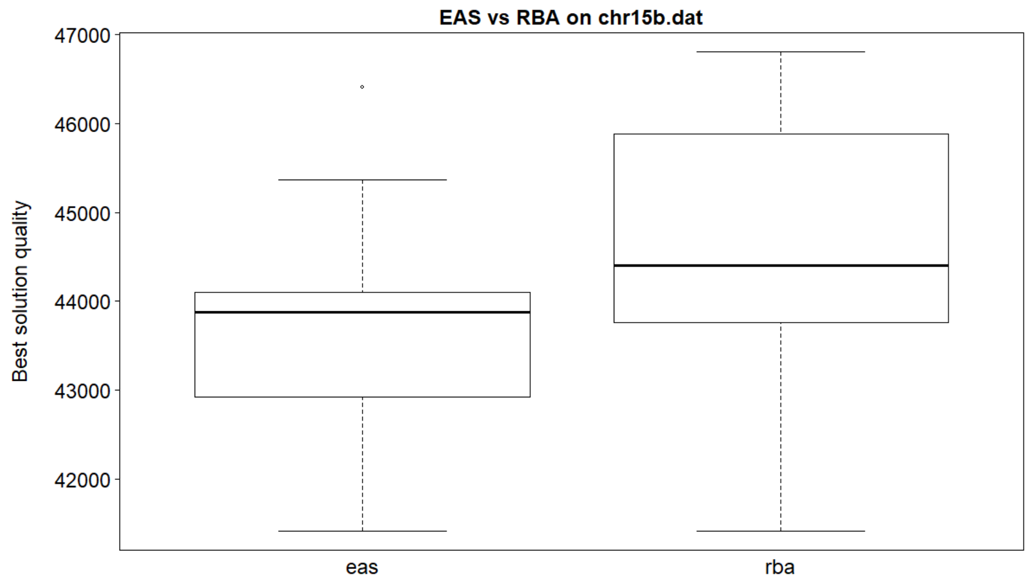

FIGURE 2 – Results on the both algorithm on the instance chr15b

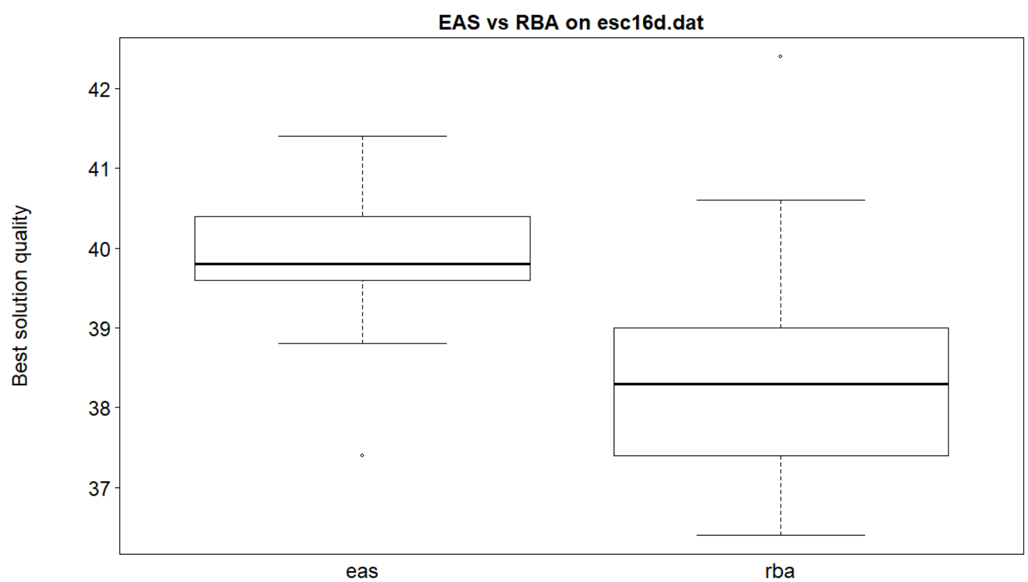

FIGURE 3 – Results on the both algorithm on the instance esc16d

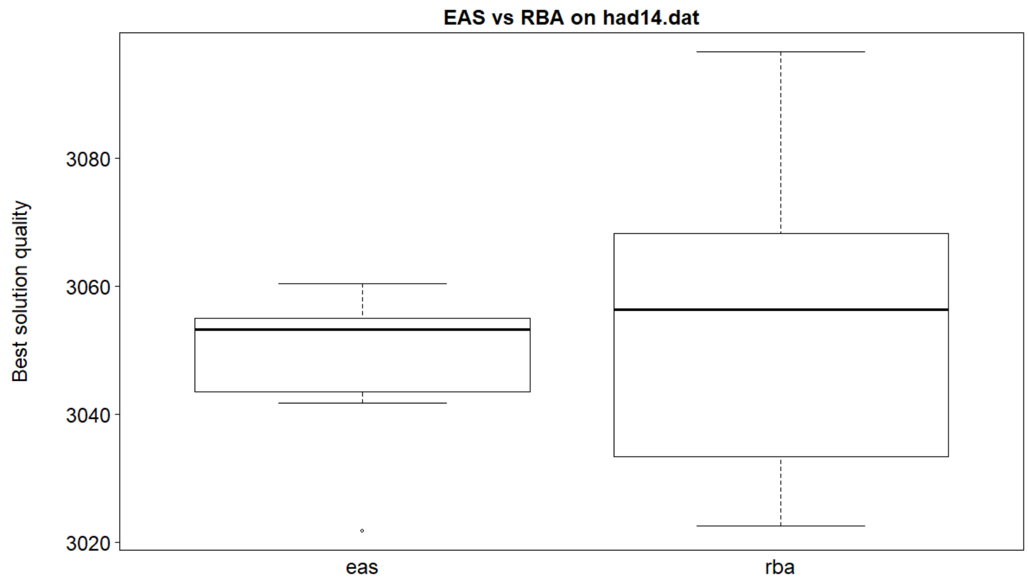

FIGURE 4 – Results on the both algorithm on the instance had14

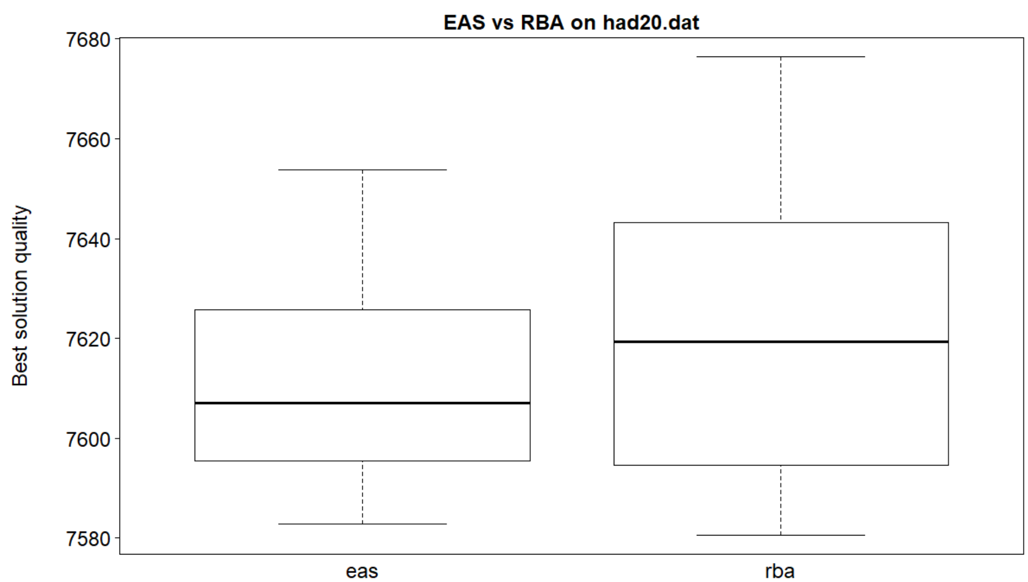

FIGURE 5 – Results on the both algorithm on the instance had20

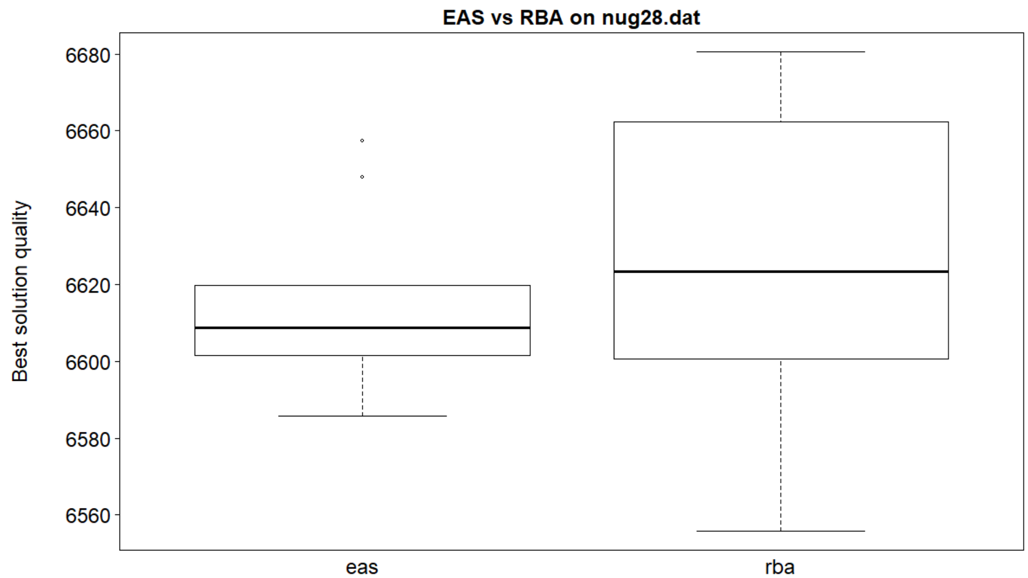

FIGURE 6 – Results on the both algorithm on the instance nug28

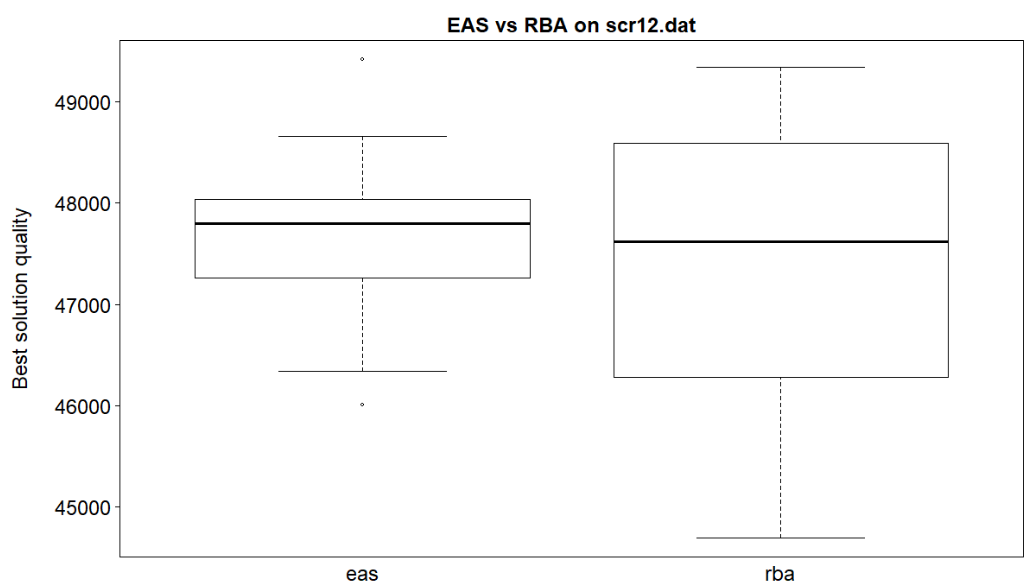

FIGURE 7 – Results on the both algorithm on the instance scr12

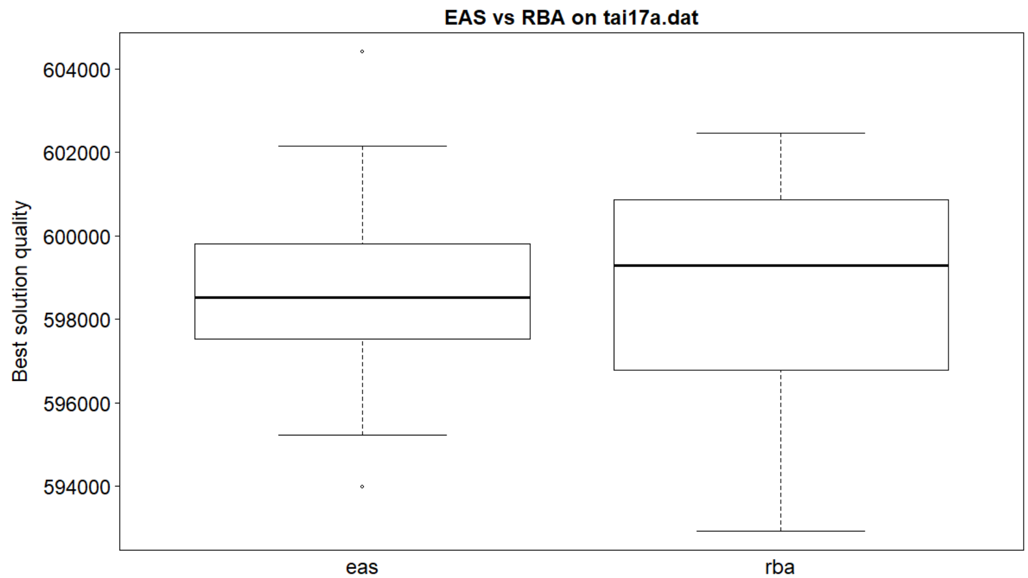

FIGURE 8 – Results on the both algorithm on the instance tai17a

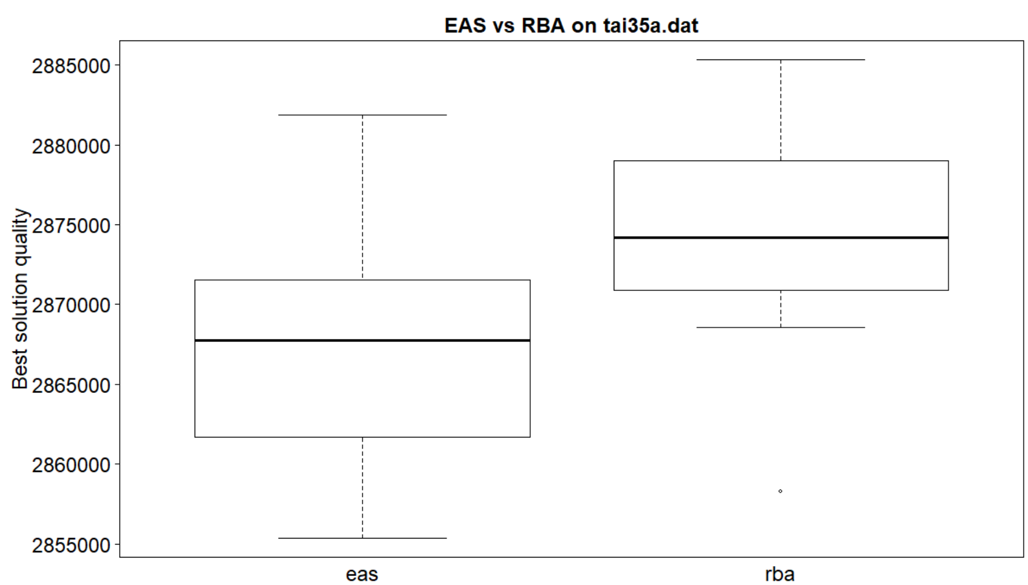

FIGURE 9 – Results on the both algorithm on the instance tai35a

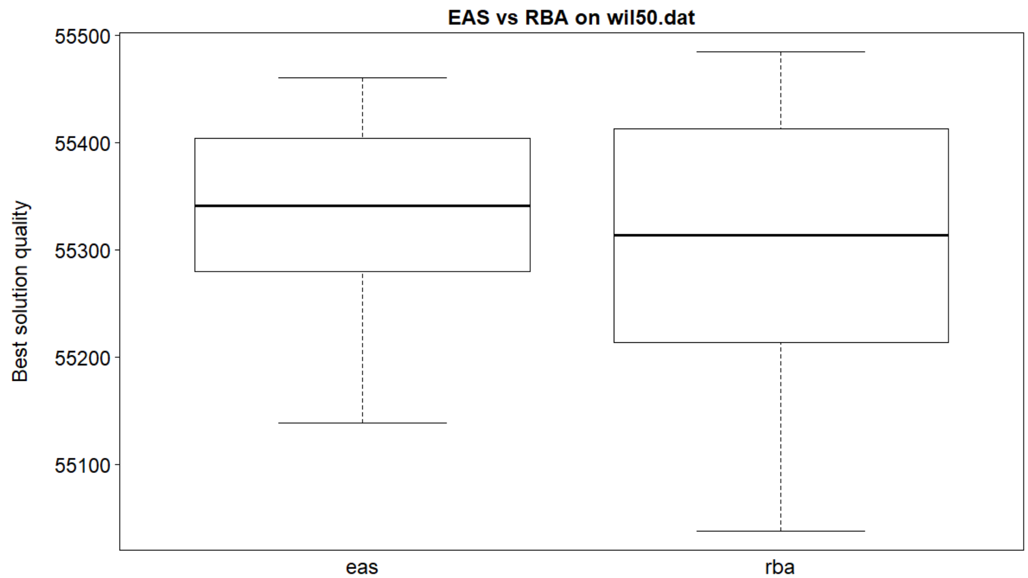

FIGURE 10 – Results on the both algorithm on the instance wil50

3.1.2 Wilcoxon

Wilcoxon signed rank (paired)

	"eas"	"rba"
"eas"	NA	0.375
"rba"	NA	NA

Bonferroni correction

	"eas"	"rba"
"eas"	NA	0.375
"rba"	NA	NA

3.1.3 Convergence

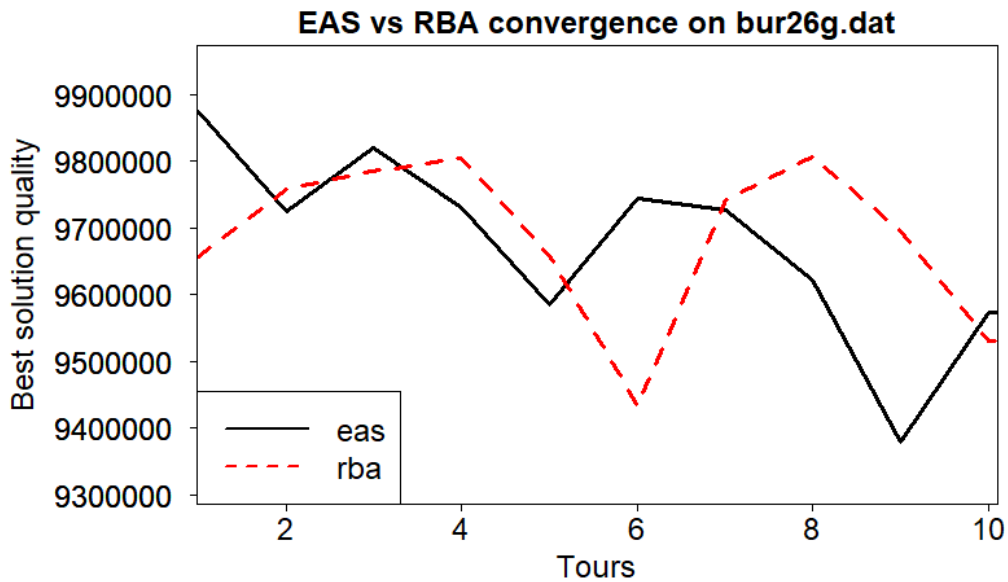

FIGURE 11 – Results on the both algorithm on the instance bur26g

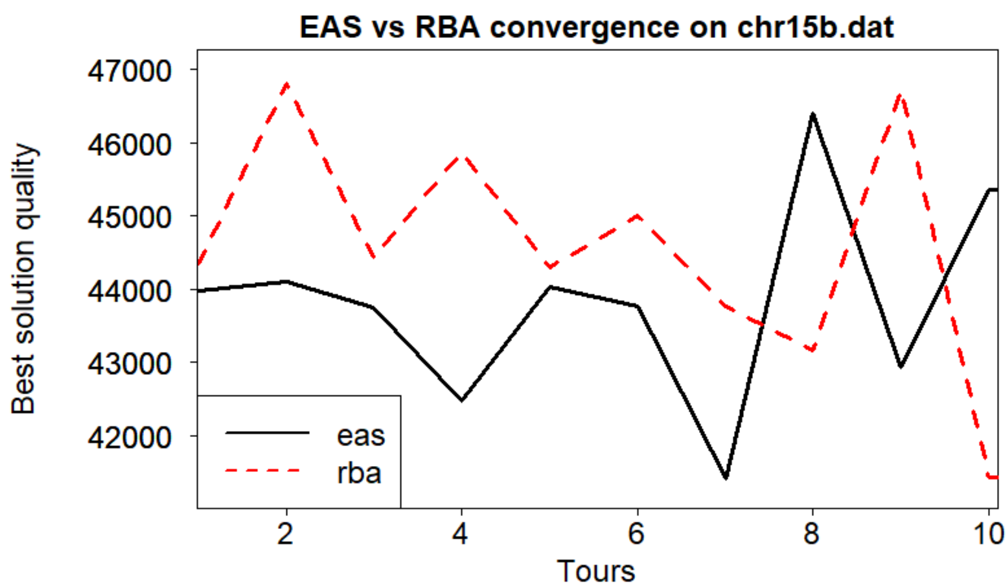

FIGURE 12 – Results on the both algorithm on the instance chr15b

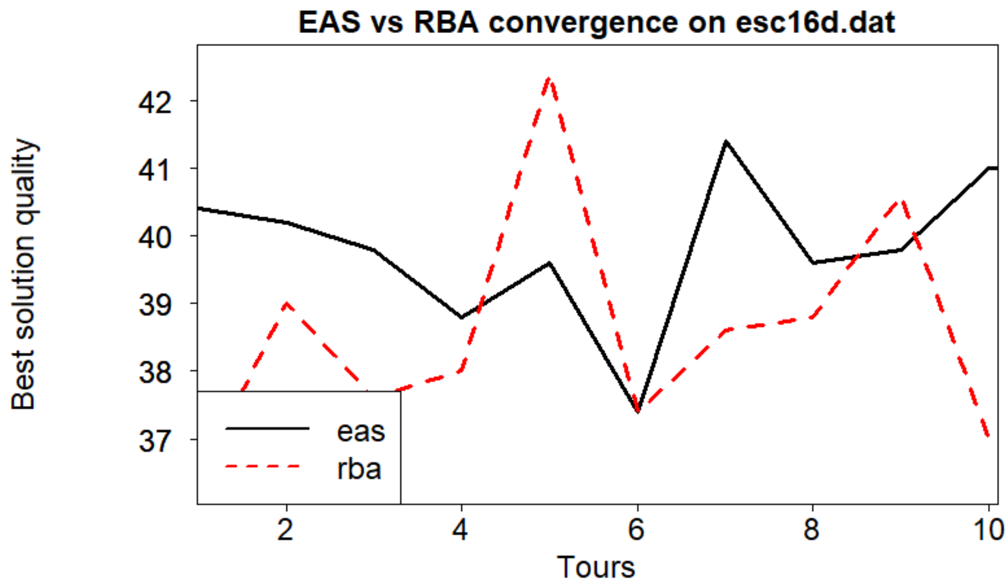

FIGURE 13 – Results on the both algorithm on the instance esc16d

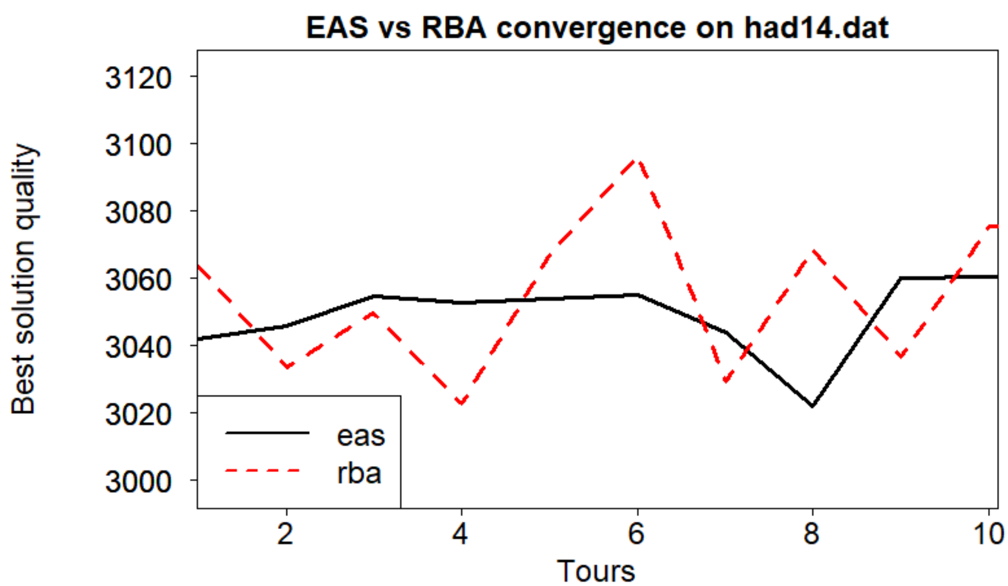

FIGURE 14 – Results on the both algorithm on the instance had14

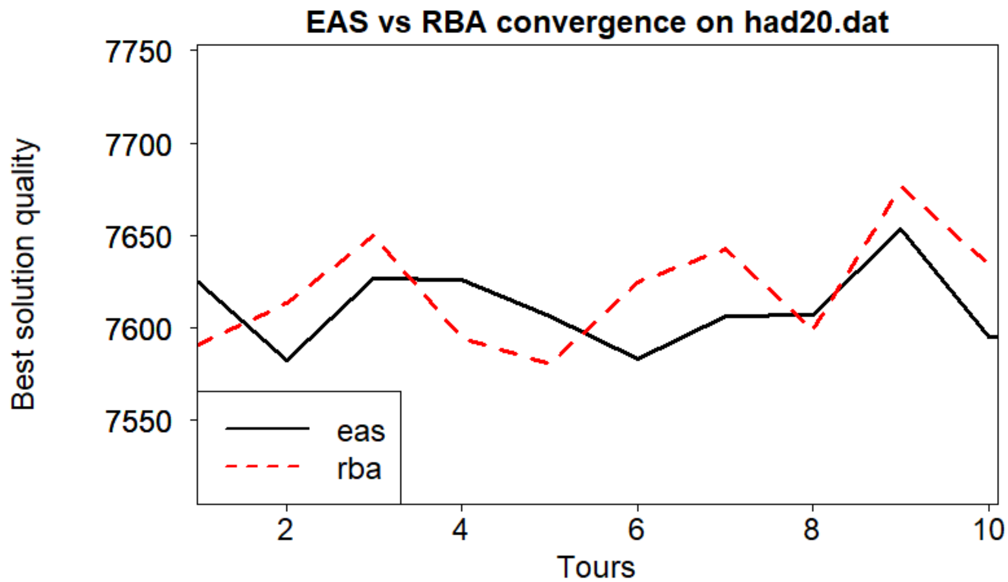

FIGURE 15 – Results on the both algorithm on the instance had20

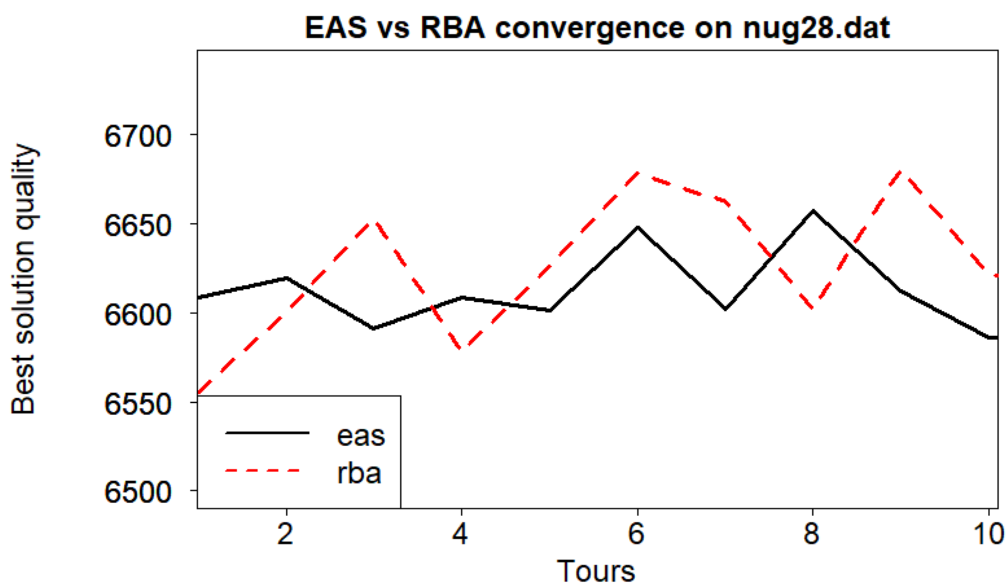

FIGURE 16 – Results on the both algorithm on the instance nug28

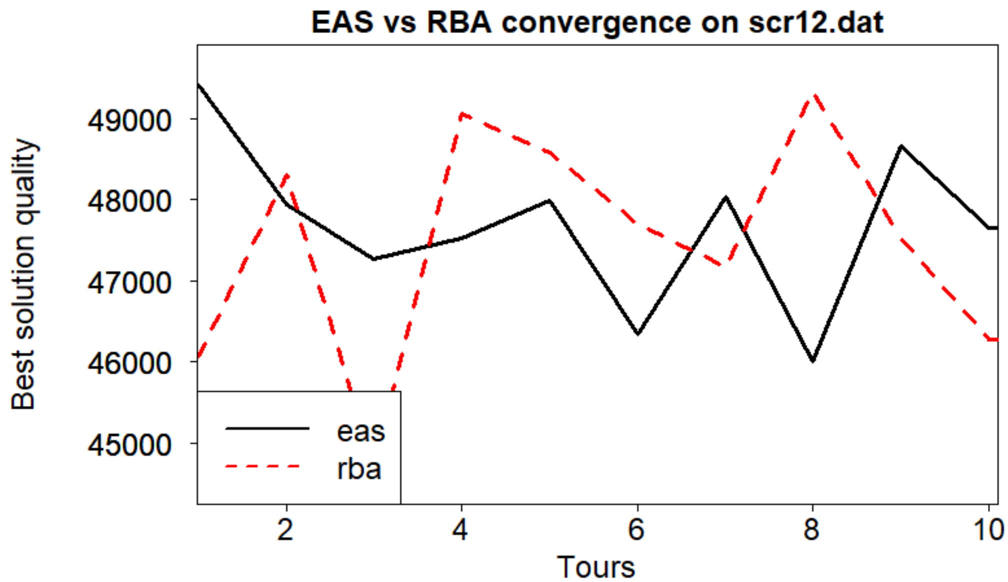

Figure 17 – Results on the both algorithm on the instance scr12

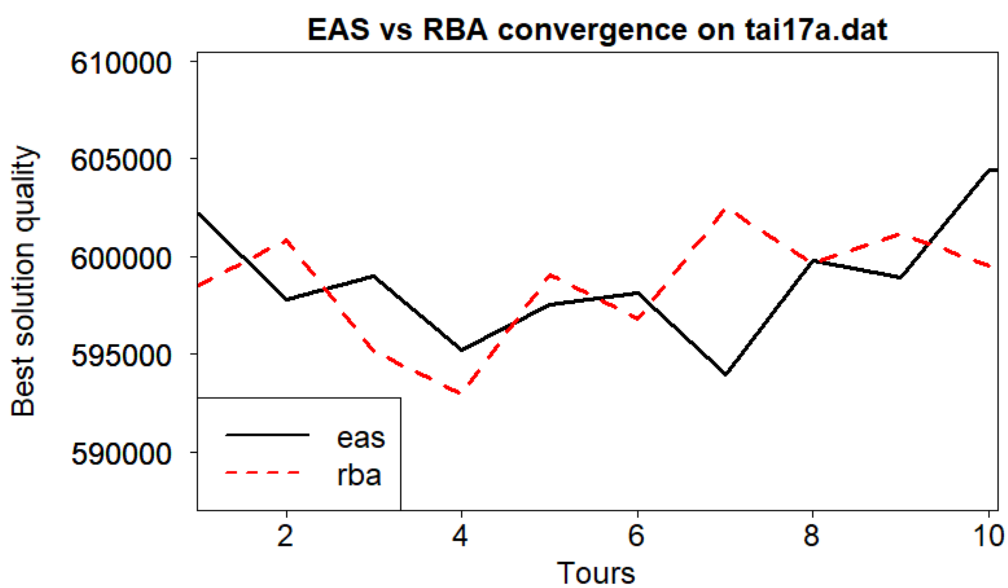

Figure 18 – Results on the both algorithm on the instance tai17a

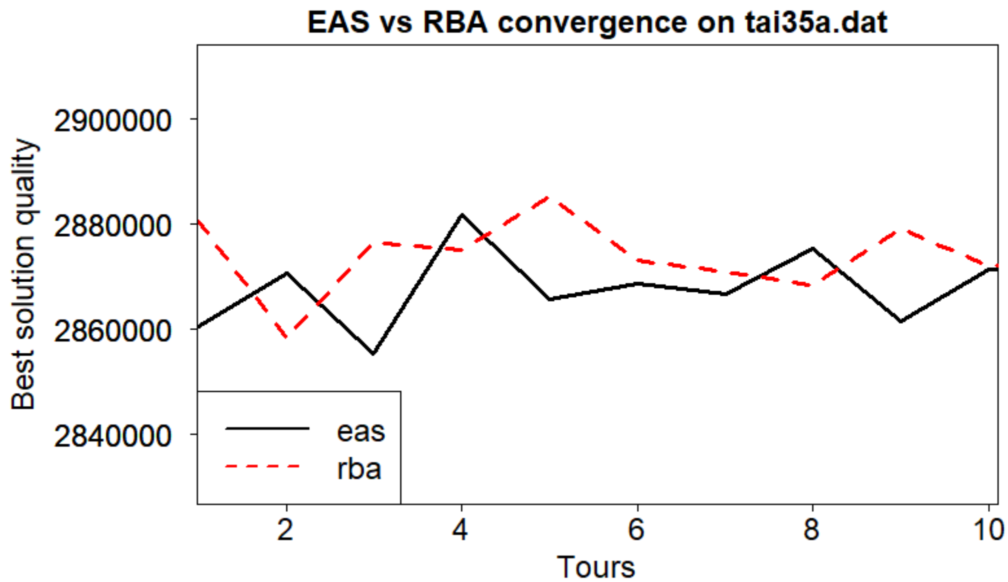

FIGURE 19 – Results on the both algorithm on the instance tai35a

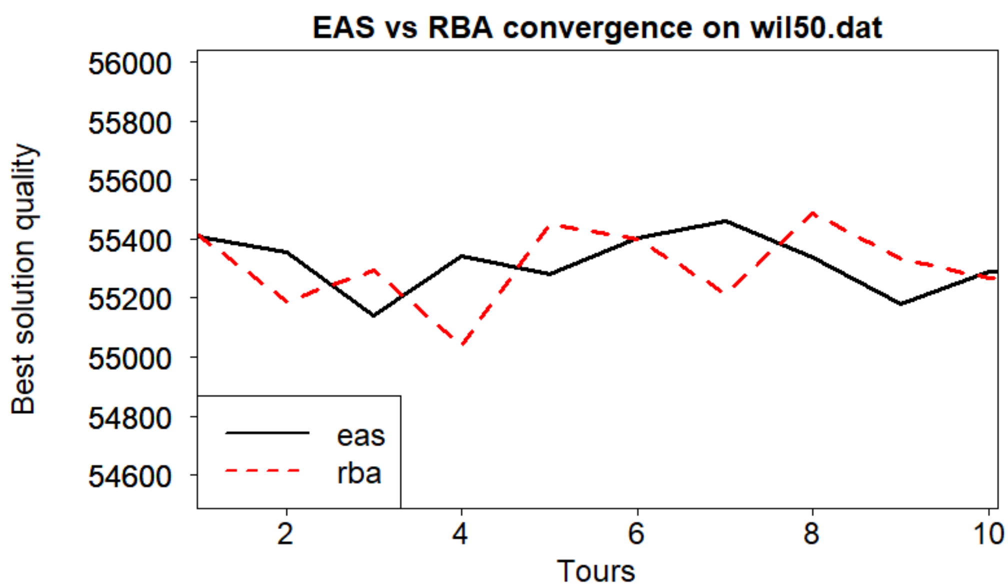

FIGURE 20 – Results on the both algorithm on the instance wil50

3.2 Analysis

As we can see in the results, we can determine that the Rank-Based Ant system, in small majority of instances, give a better results than the Elitist Ant System. With the Wilcoxon test ranked, there is a little change in the median for both algorithm in favor of the Rank-Based Ant system. The corrected Bonferroni is under 1, so the Rank-Based Ant is a little bit better than the Elitist Ant system. About the convergence, we can notice that both algorithms are noisy in some instances but the behavior of both is almost the same. In a multiple experiments, I noticed that, sometimes, EAS is better than RBA but in general, the RBA returns a best results in the most of instances. In majority, RBA returns a best solution but not necessarily a better performance.

So we can conclude that the **Rank-Based Ant** algorithm is recommend for solving the Quadratic Assignment Problem.

4 RBA vs local search

In this section, we will compare the performance between the **RBA** and his local search version. We will reproduce the same experiments with the same set of seeds mentioned in the section 3. The local search applied, in this case, is the 2-opt algorithm. The 2-opt is algorithm in which we permute 2 edges and compare the results. If the results after the permutation is better than before, the permutation is kept. In the case of Ant Colonization, When an ant finish to build his solution, we loop over his tour and we permute 2 cities. If the permutation give a better cost, we save the permutation and we continue until we reach the end of the array ($\Theta(N)$ where N is the number of city).

4.1 Results

4.1.1 Best solution quality

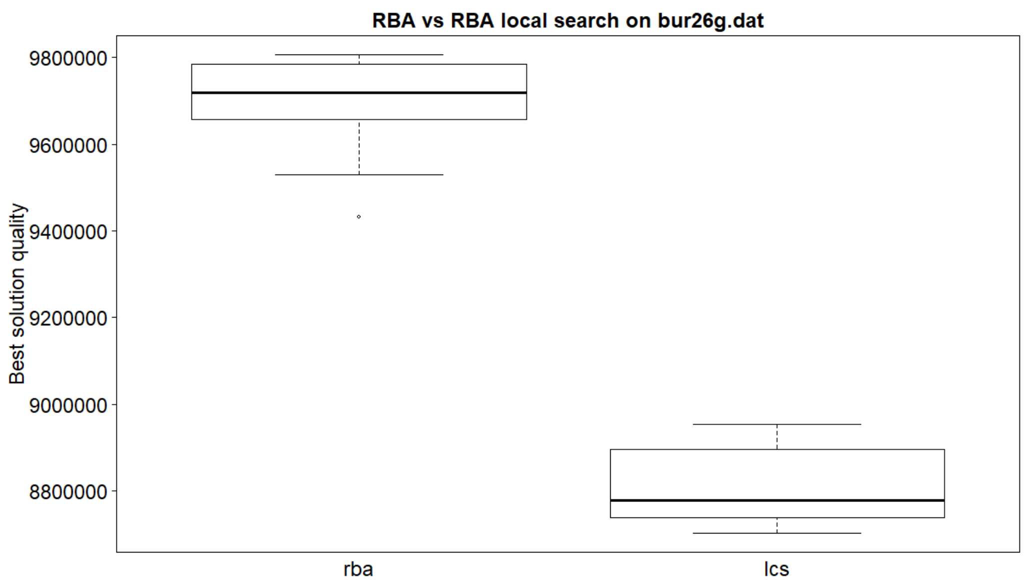

FIGURE 21 – Results on the both algorithm on the instance bur26g

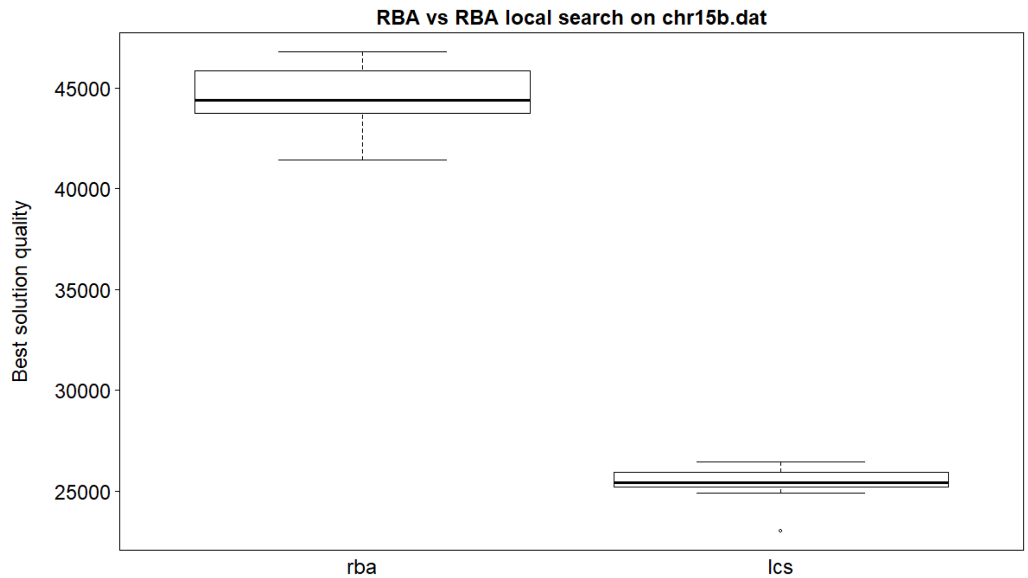

FIGURE 22 – Results on the both algorithm on the instance chr15b

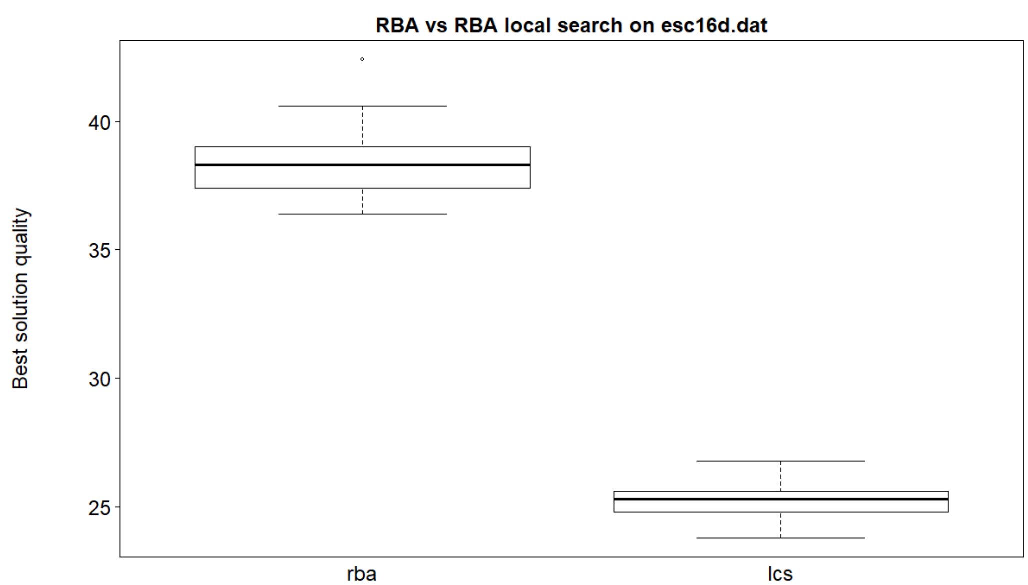

FIGURE 23 – Results on the both algorithm on the instance esc16d

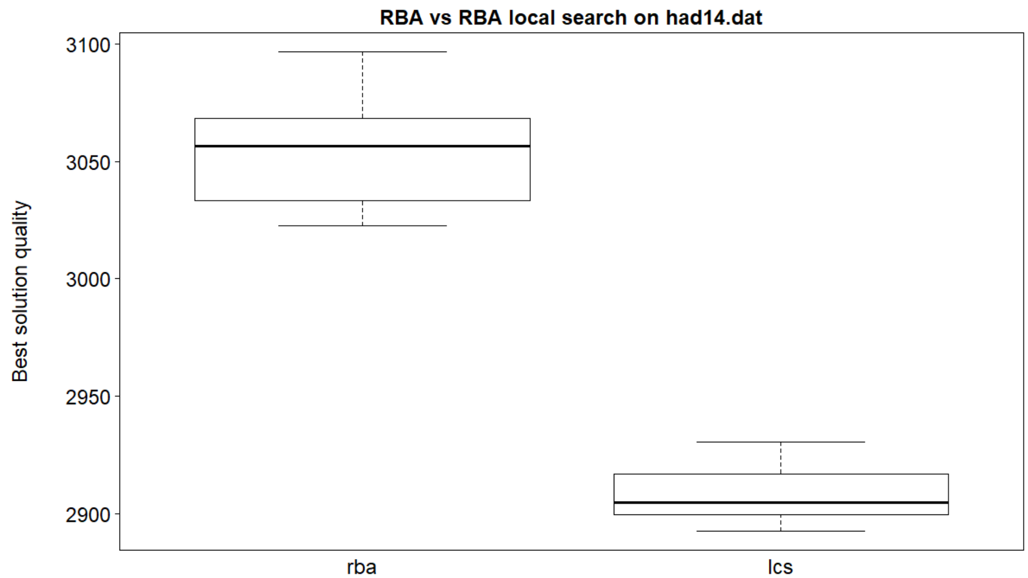

FIGURE 24 – Results on the both algorithm on the instance had14

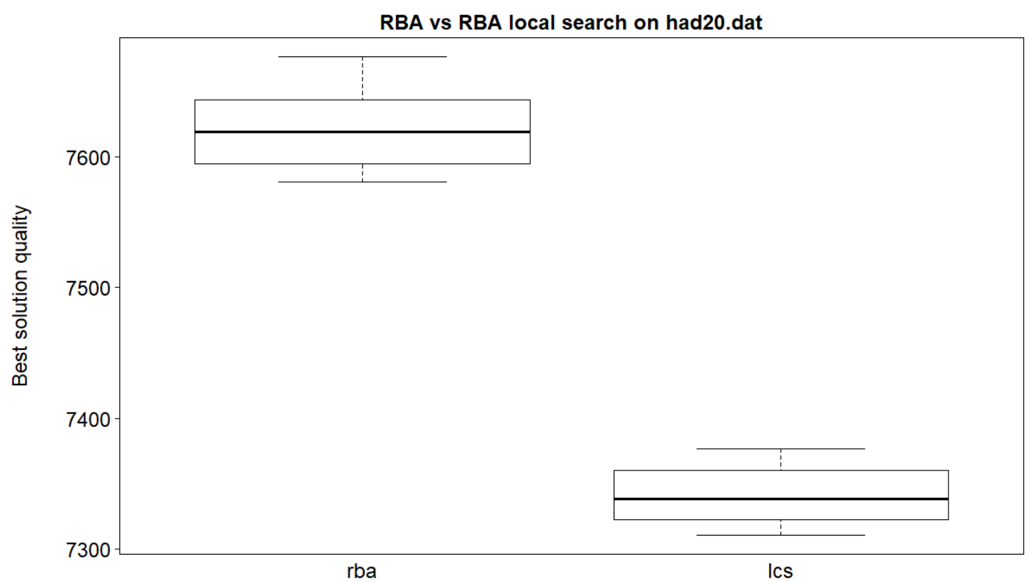

FIGURE 25 – Results on the both algorithm on the instance had20

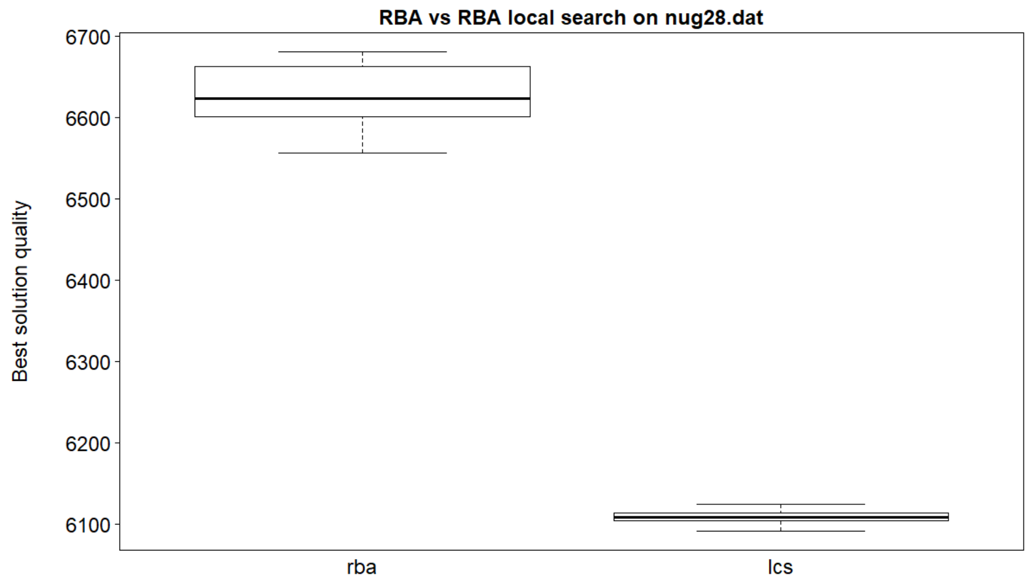

FIGURE 26 – Results on the both algorithm on the instance nug28

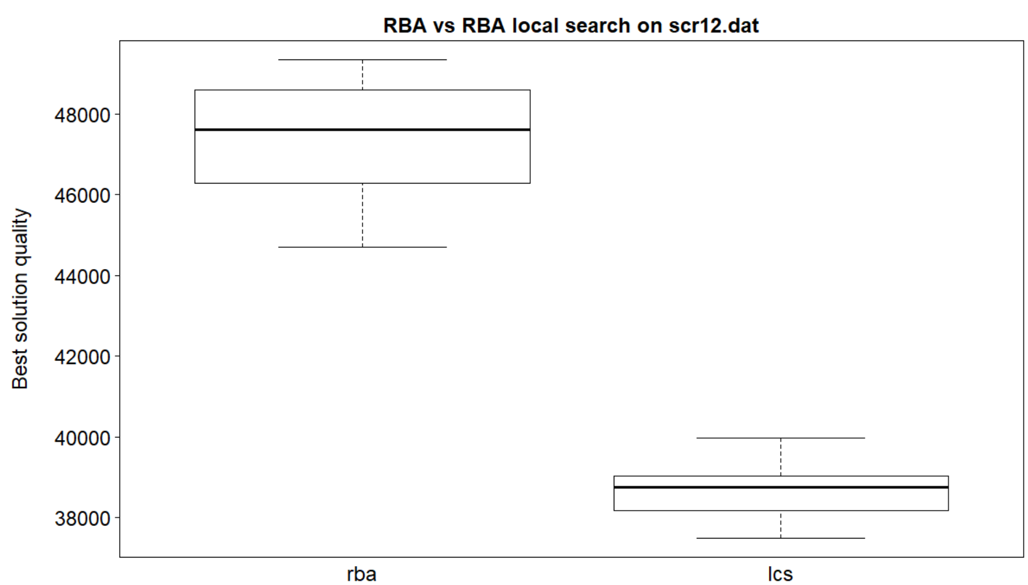

FIGURE 27 – Results on the both algorithm on the instance scr12

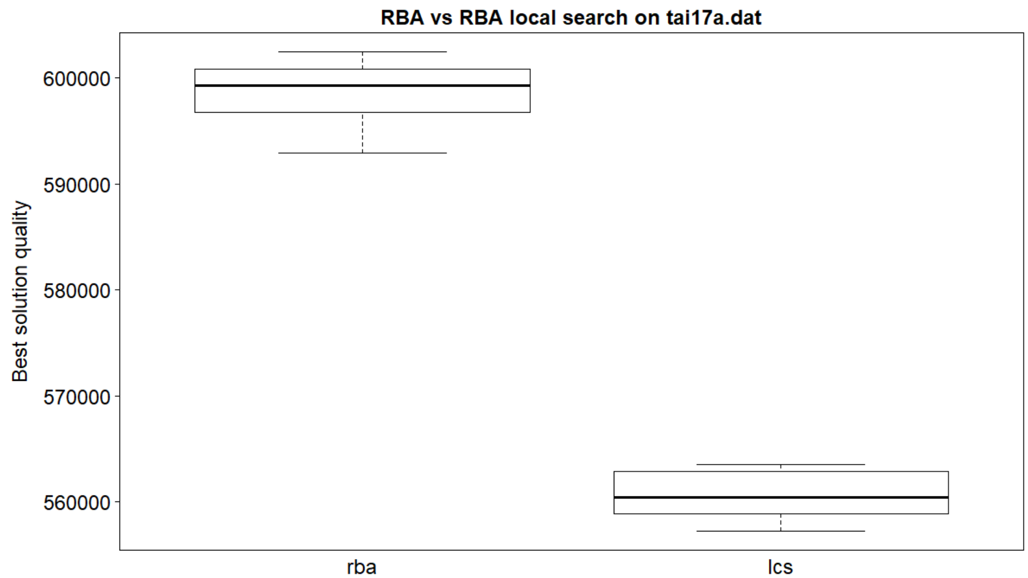

FIGURE 28 – Results on the both algorithm on the instance tai17a

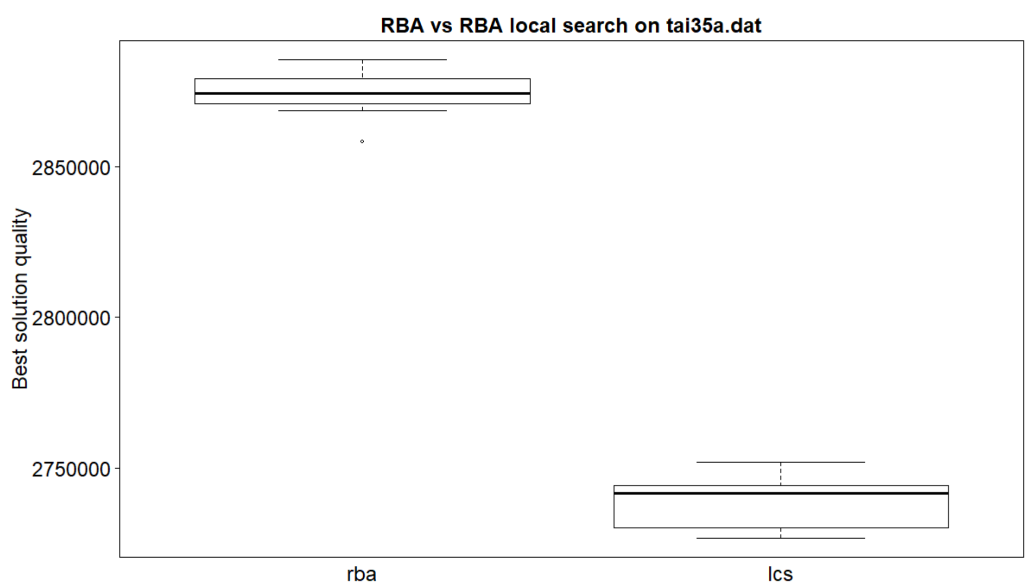

FIGURE 29 – Results on the both algorithm on the instance tai35a

4.1.2 Wilcoxon test ranked

Wilcoxon signed rank (paired)				Bonferroni correction	
	"rba"	"lcs"		"rba"	"lcs"
"rba"	NA	0.001953125	"rba"	NA	0.001953125
"lcs"	NA	NA	"lcs"	NA	NA

4.1.3 Convergence

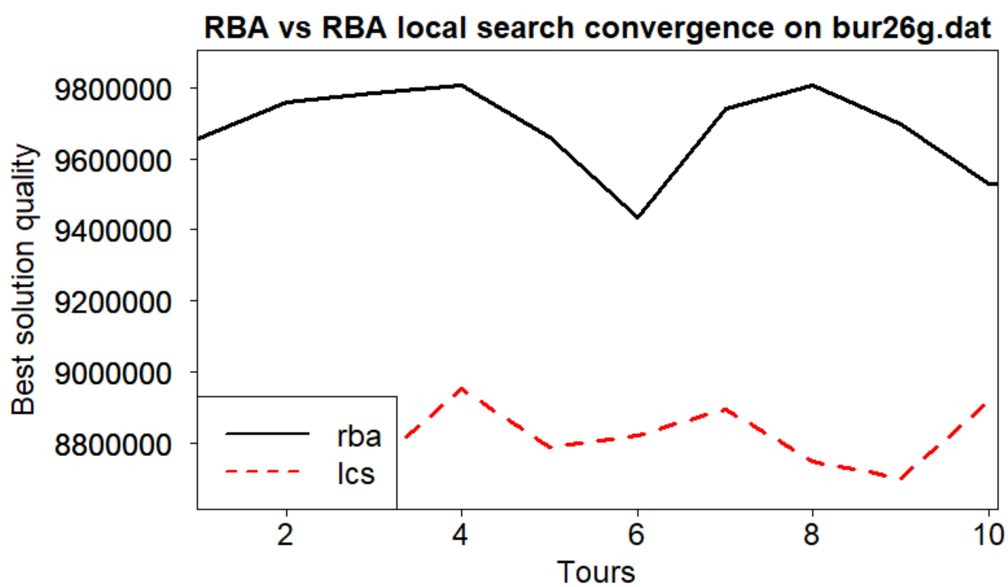

FIGURE 30 – Results on the both algorithm on the instance bur26g

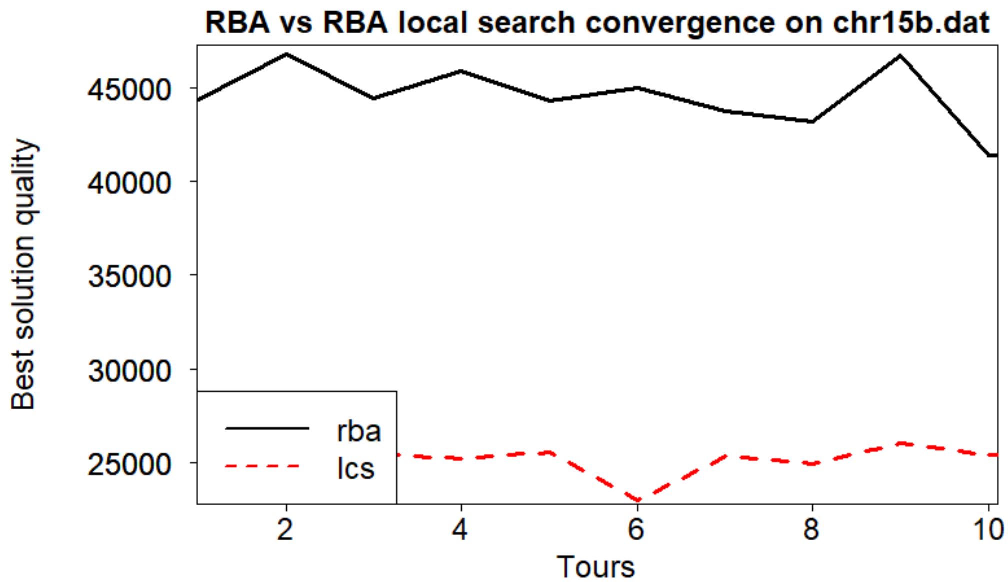

FIGURE 31 – Results on the both algorithm on the instance chr15b

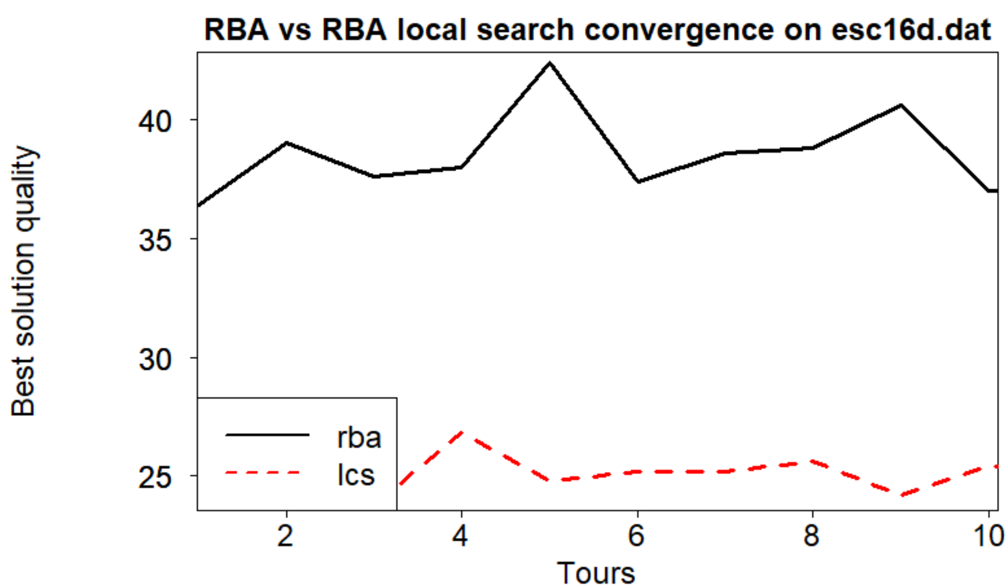

FIGURE 32 – Results on the both algorithm on the instance esc16d

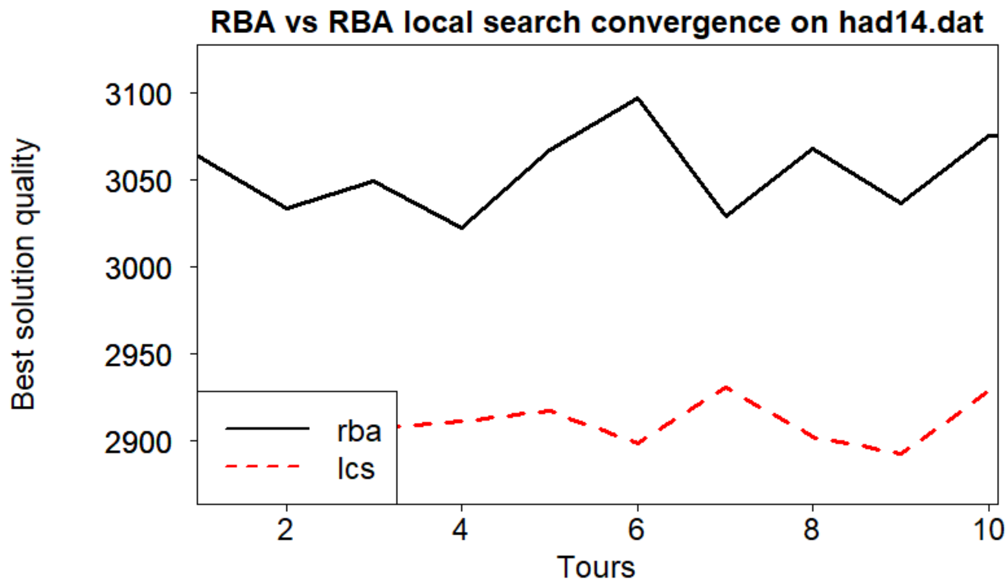

FIGURE 33 – Results on the both algorithm on the instance had14

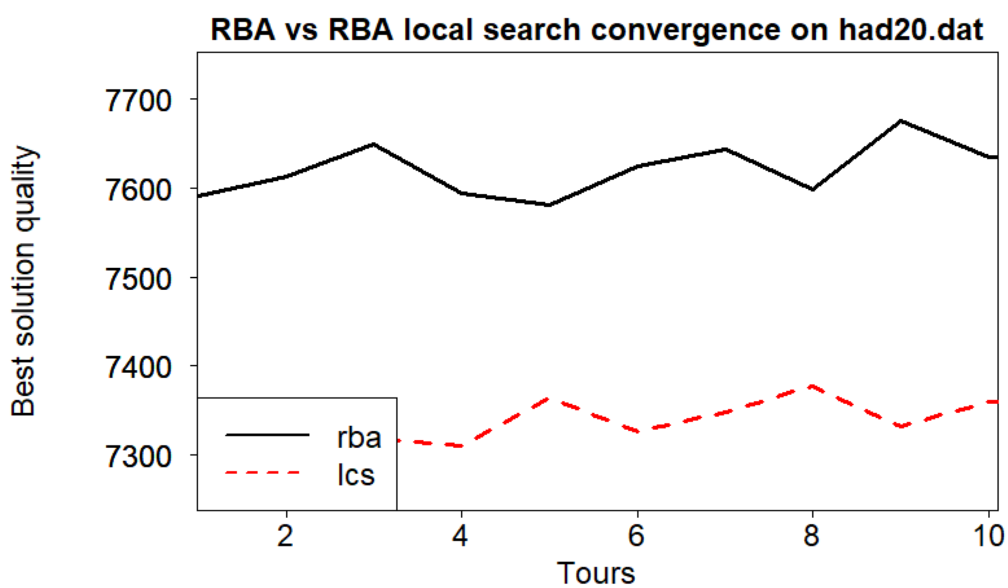

FIGURE 34 – Results on the both algorithm on the instance had20

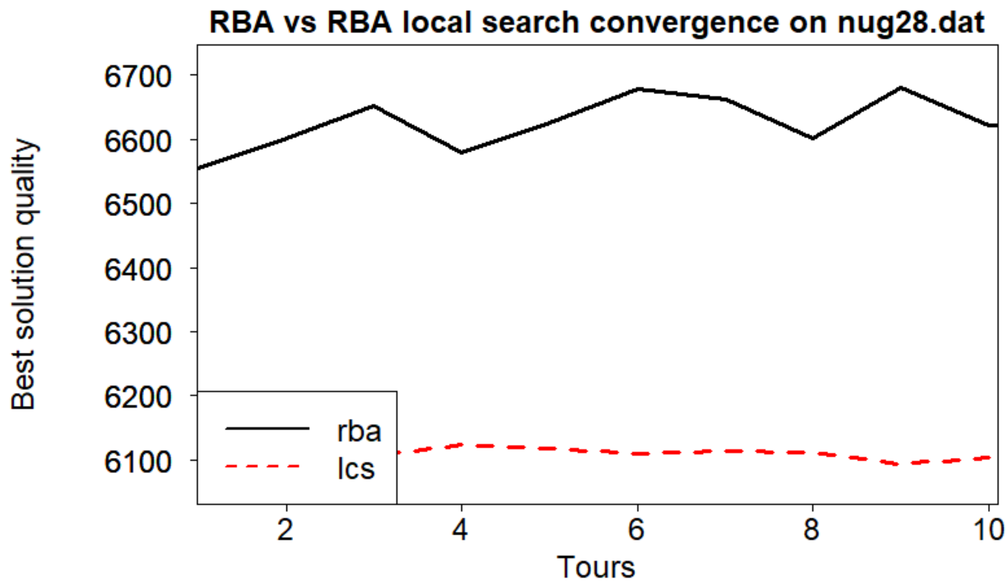

FIGURE 35 – Results on the both algorithm on the instance nug28

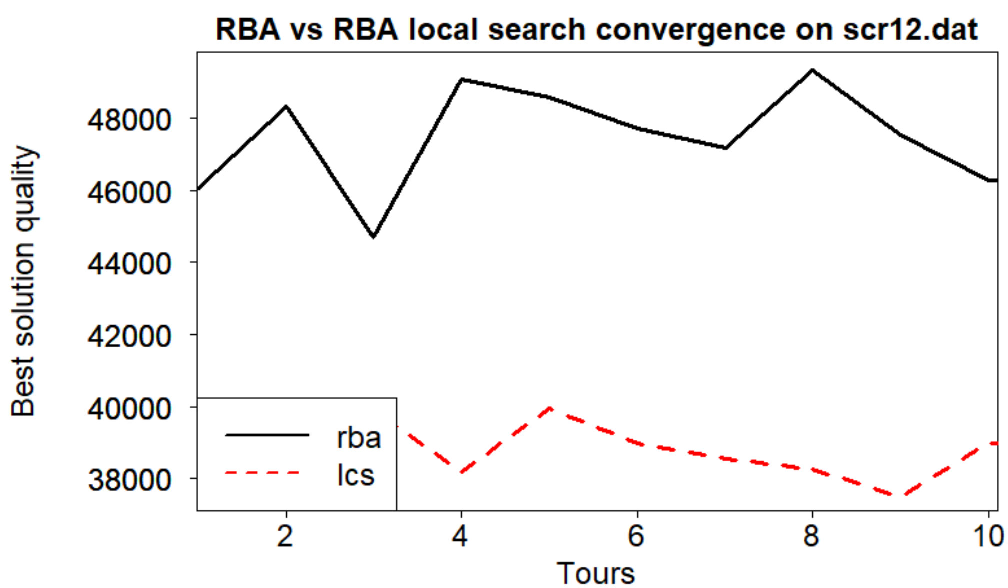

FIGURE 36 – Results on the both algorithm on the instance scr12

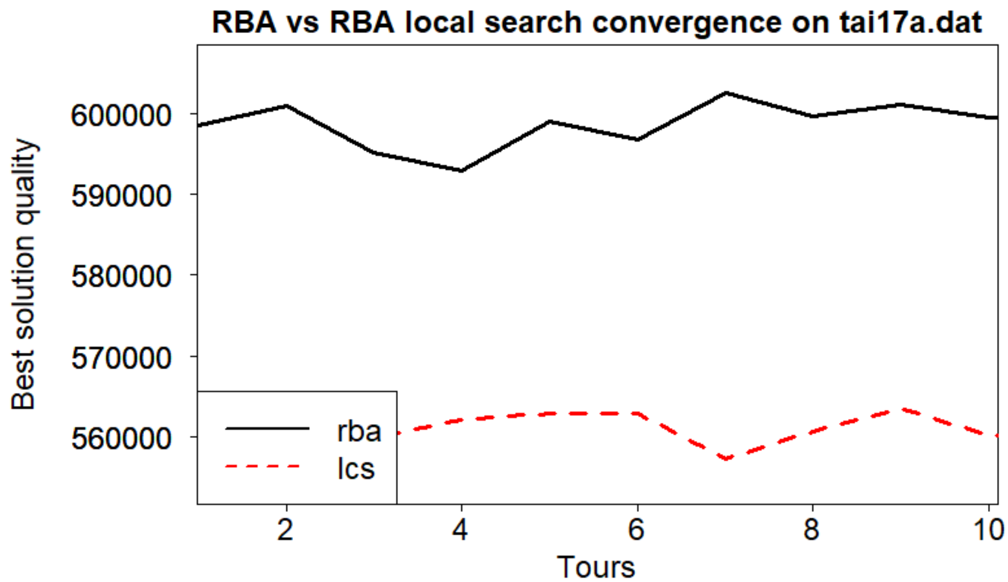

FIGURE 37 – Results on the both algorithm on the instance tai17a

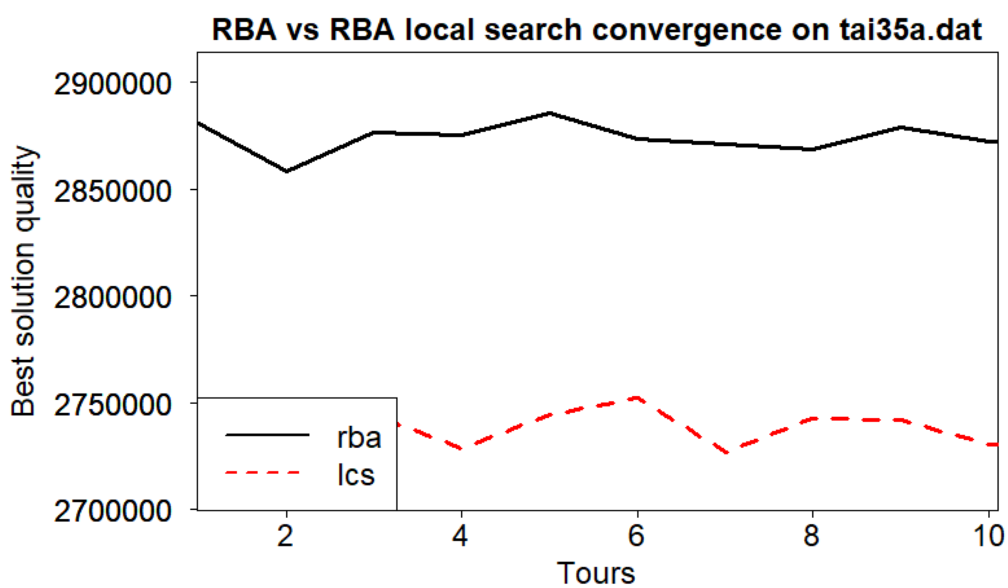

FIGURE 38 – Results on the both algorithm on the instance tai35a

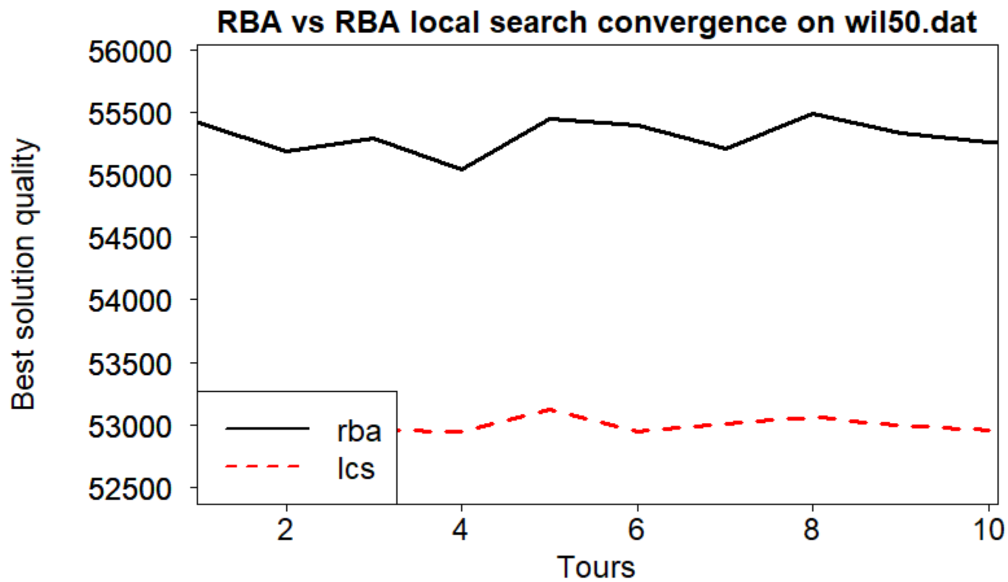

FIGURE 39 – Results on the both algorithm on the instance wil50

4.2 Analysis

As we can expect, the local search provide a much more better results than the version without local search. The Rank-Based Ant system still provide good results compared to the Elitist Ant but the RBA system with local search returns a much more better result. When we tune the parameters of the algorithms, the results of RBA with local search don't have a really impact. The results and the quality remains almost the same, only the behavior of the curve change a little bit but doesn't give a major difference.

5 Conclusion

We compared the Elitist Ant and Rank-based Ant for seeing which one give the best results on the Quadratic Assignment Problem. We noticed that both algorithm are almost the same. Sometimes, EAS will provide a better solution than RBA or reverse. In a most cases, RBA returns the best one. About performance, both algorithm has the same one. Based on the experiments, we can conclude that the best (of both) algorithm for solving the QAP is *Rank-Based Ant system*.