Companion to Off-line and On-line Tuning: A Study on Operator Selection for a Memetic Algorithm Applied to the QAP

Gianpiero Francesca[1], Paola Pellegrini[2],
Thomas Stützle[2], and Mauro Birattari[2]

[1] Dipartimento di Ingegneria, University of Sannio, Benevento, Italy
gianpiero.francesca@gmail.com
[2] IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
paolap@pellegrini.it; stuetzle@ulb.ac.be; mbiro@ulb.ac.be

In the paper [1] we study the performance of three memetic algorithms when their configuration is tuned either off-line or on-line. We assess the relative performance achieved by algorithms of different quality tuned either off-line or on-line. We tackle the quadratic assignment problem. The results suggest that off-line tuning achieves in general better performance than on-line tuning. On-line is either comparable or more profitable than off-line tuning only if the quality of the algorithm is rather poor. Here we present the results of the whole experimental analysis, on three memetic algoritms (MA) and on two $\mathcal{MAX}$–$\mathcal{MIN}$ ant system ($\mathcal{MM}$AS) .

1 Experimental setup

In the experimental analysis we study the performance of two $\mathcal{MM}$AS and three MA algorithms when the configuration to be used is selected through either an off-line or an on-line tuning method. By studying the various $\mathcal{MM}$AS and MA algorithms, we compare the performance achieved by the different tuning methods as a function of the quality of the algorithm. The three memetic algorithms follow the implementation proposed by Merz and Freisleben [2]. They differ in the application of either the local search or the mutation operator. In the first algorithm (simple MA) we implement neither the local search nor the mutation operator. In the second algorithm (intermediate MA) we implement the local search, while we apply no mutation operator. In the third algorithm (complex MA) we implement both the local search and the mutation operator. The two $\mathcal{MM}$AS algorithms are implementation of $\mathcal{MAX}$–$\mathcal{MIN}$ ant system ($\mathcal{MM}$AS) for the QAP described by Stützle and Hoos [3]. In the first algorithm (simple $\mathcal{MM}$AS) we does not implement the local search. In the second algorithm (complex $\mathcal{MM}$AS) we implement the local search.

We analyze the performance of the algorithms when different values of CPU time are imposed as stopping criterion. We run the algorithms with the following parameter settings: in $\mathcal{MM}$AS $m = 2$, $\rho = 0.6$; in MA $p = 40$, $p_c = p/2$, $t = 30\%$; in complex MA $p_m = p/2$, $m = 30\%$. In all the cases we apply the 2-opt local search with best improvement [3].

Table 1. Configuration selected by off-line tuning for MA on the two sets of instances tackled, for each algorithm and for each CPU time considered as stopping criterion [s].

	heterogeneous			homogeneous		
time	10	31	100	10	31	100
simple	PMX	PMX	PMX	OX	PMX	PMX
intermediate	CX	PMX	PMX	CX	CX	CX
complex	CX	PMX	PMX	CX	PMX	PMX

For each algorithm, we consider seven different versions depending on the configuration selection policy:

- The configuration is maintained constant through the whole run: the configuration to be used is i) *literature*, L: the one suggested in the literature [4, 5], $\alpha = 1$ in $\mathcal{MM}$AS and crossover operator CX in MA; ii) *off-line*, OFF: the one selected by the off-line method; *random*, R: randomly selected according to a uniform probability distribution.
- The configuration to be used is selected at each step: iv) *naive*, N: random selection of the configuration, according to a uniform probability distribution. In $\mathcal{MM}$AS v) *local search*, LS; vi) self-adaptive with parameter managed at the colony level, SAc; vii) self-adaptive with parameter managed at the ant level, SAa. In MA v) *probability matching* with $\beta = 0.3$ and $P_{min} = 0.05$, PM; vi) *adaptive pursuit* with $\beta = 0.3$,$\gamma = 0.3$ and $P_{min} = 0.05$, AP; vii) *multi-armed bandit* with $\delta = 1$.

The naive and random versions are used as benchmarks.

For a fair comparison between off-line and on-line tuning, all the methods select the configuration to be used from the same set of possibilities: for $\mathcal{MM}$AS, $\alpha \in (0.5, 1, 1.5, 2, 3)$; for MA, crossover operator $\in$(CX, DPX, PMX, OX).

We consider two sets of instances, named *heterogeneous* and *homogeneous*, respectively. The heterogeneous set contains 34 instances from the QAPLIB [6]. We tackle all the instances of size 50 to 100. The homogeneous set contains 34 unstructured instances of size 80, obtained through the instance generator described by Stützle and Fernandes [7]. The two sets of instances are randomly split in two subsets. One of them is used for performing the off-line tuning. Table 1 reports the configuration selected by off-line tuning for each algorithm and for each value of CPU time imposed as stopping criterion, namely 10, 31 and 100 seconds. We fixed these values following a multiplicative scale of step $\sqrt{10}$.

In section 2 we discuss the results achieved on the instances of the second subsets by the seven versions implemented. For the different stopping criteria, we perform 10 independent runs of each version on all instances.

All the experiments are performed on Xeon E5410 quad core 2.33GHz processors with 2x6 MB L2-Cache and 8 GB RAM, running under the Linux Rocks Cluster Distribution. The algorithms are implemented in C++, and the code is compiled using gcc 4.1.2.

Table 2. Algorithm quality. Percentage of error obtained after 100 seconds by the literature version of the three MA algorithms, with respect to the best known solution of each instance.

	homogeneous	heterogeneous
simple	4.69343 %	9.29772 %
intermediate	1.51216 %	2.17695 %
complex	0.79046 %	1.44571 %

2 Experimental results

By analyzing the performance of three MA and two $\mathcal{MM}$AS algorithms, we observe the impact of the tuning method, either off-line or on-line, as a function of the quality of the algorithm.

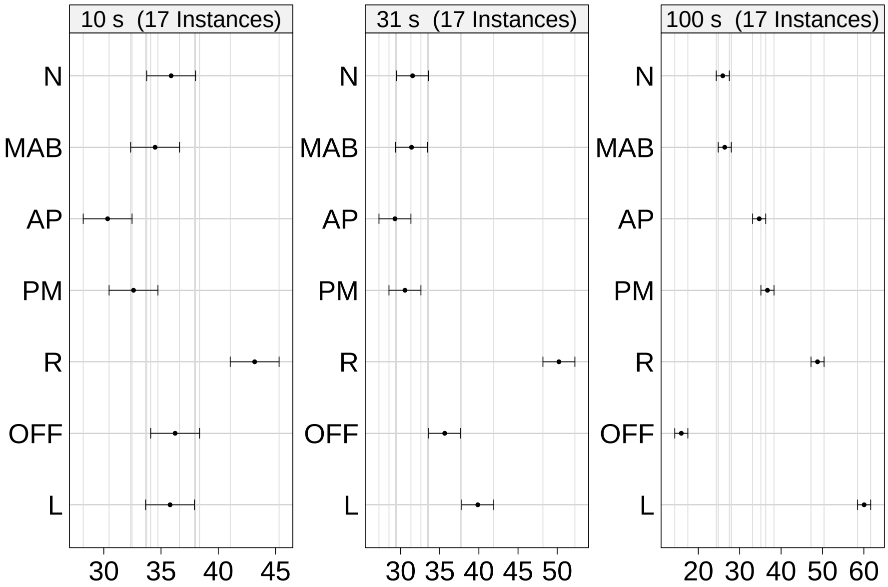
2.1 Memetic Algorithms MA

Table 1 reports the configuration selected by the off-line tuning for the three versions of the MA for different computational run-lengths. Table 2 shows the percentage error obtained by the literature version of the three algorithms in one run of 100 seconds on the homogeneous and heterogeneous instances excluded by the tuning phase, with respect to the best known solution of each instance. The best algorithm is the complex one, followed by the intermediate. The simple algorithm is the worst performing. For each instance set the difference between all pairs of algorithms is statistically significant at the 95% confidence level, according to the Wilcoxon rank-sum test.

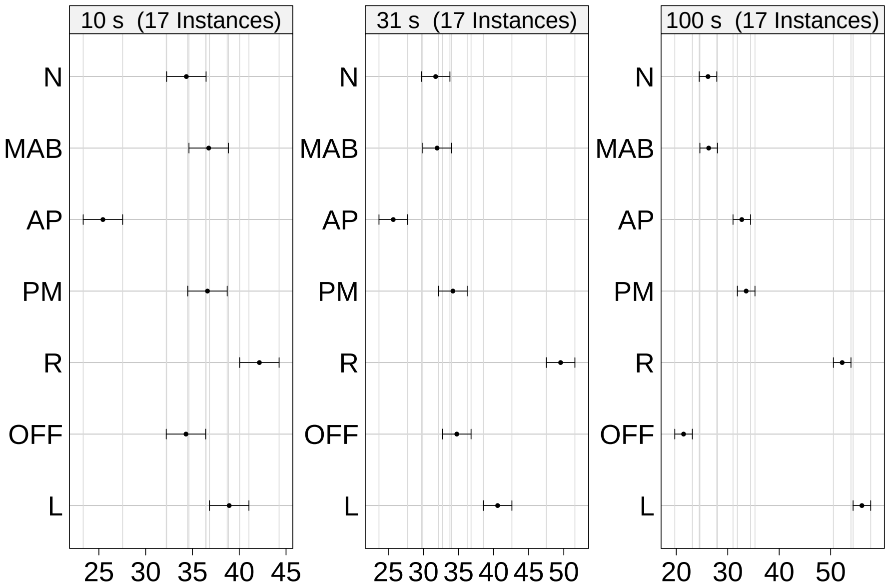
By examining the results obtained by the three different MA algorithm, we cannot identify a clear relation between the quality of the algorithms and the relative performance of off-line and on-line tuning: off-line tuning achieves quite constantly very good performance, and thus it appears the most advantageous and conservative choice. Nonetheless, when a low quality algorithm is to be used, applying an on-line method may be preferable for short run-lengths. One critical issue in this case is the selection of the best on-line tuning method: on one hand, adaptive pursuit achieves always quite good results; on the other hand, in several cases it is not the best performing method. A further element that cannot be neglected is the good performance achieved by the naive version, compared to the more advanced methods proposed in the literature. The very small difference in the performance suggests that, in the experimental conditions set here, the main advantage offered by on-line tuning might be ascribed to the introduction of some perturbation in the configuration to be used, rather than to the particular method implemented. The heterogeneity of the set of instances to be tackled does not have a remarkable impact on the results.

2.2 Ant colony optimization

We perform the same analysis considering two $\mathcal{MM}$AS algorithms, namely $\mathcal{MAX}$–$\mathcal{MIN}$ ant system ($\mathcal{MM}$AS) for the QAP either with or without local

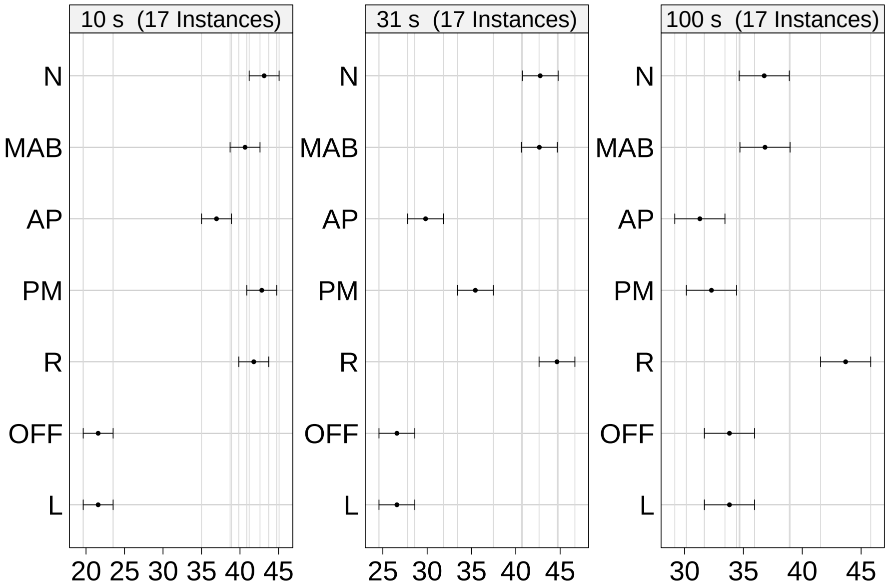


(a) homogeneous set

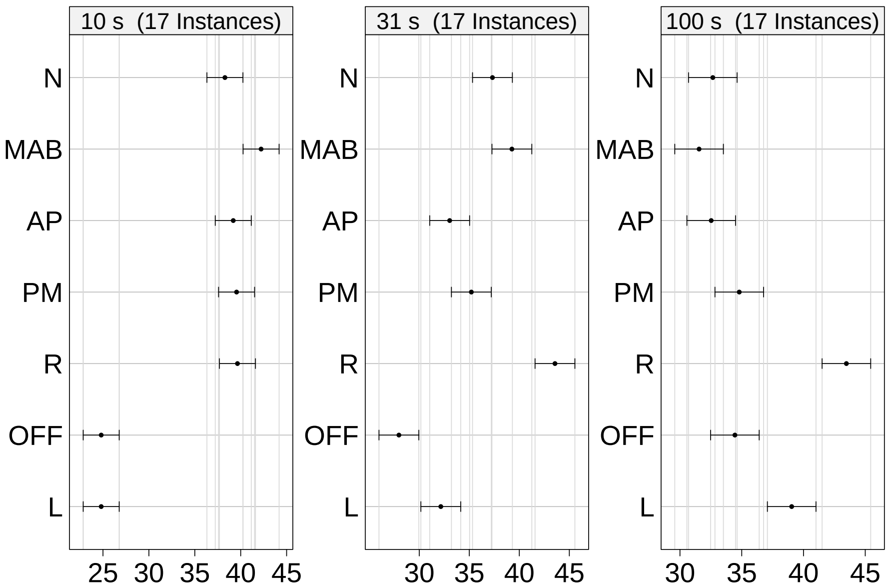


(b) heterogeneous set

Fig. 1: **Results achieved by the seven versions of the simple MA algorithm.** Simultaneous confidence intervals for all-pairwise comparisons of ranks between all versions applied to homogeneous and heterogeneous instances.

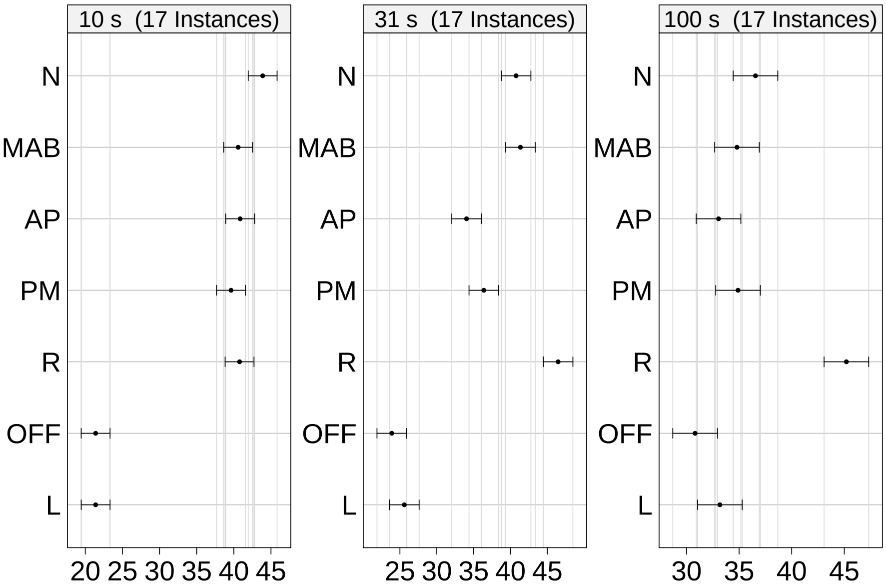


(a) homogeneous set

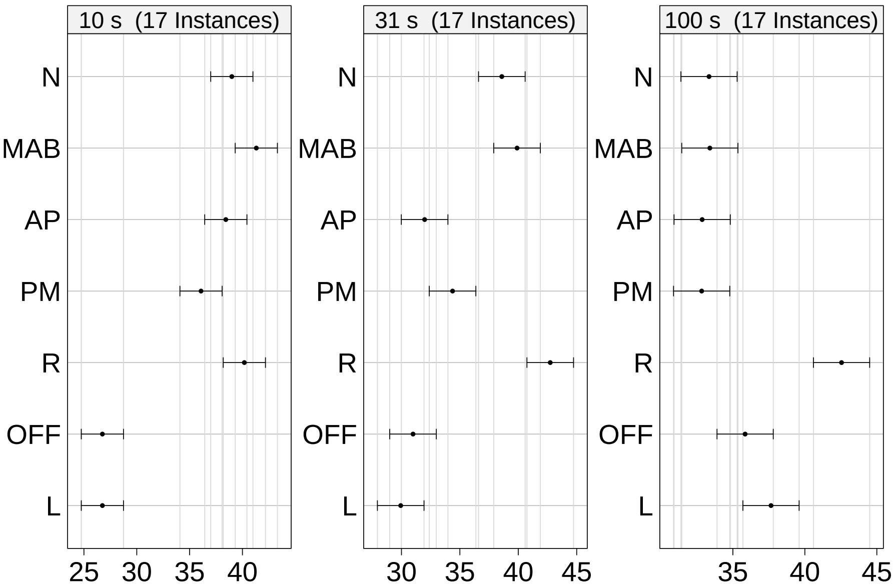


(b) heterogeneous set

Fig. 2: **Results achieved by the seven versions of the intermediate MA algorithm.** Simultaneous confidence intervals for all-pairwise comparisons of ranks between all versions applied to homogeneous and heterogeneous instances.



(a) homogeneous set



(b) heterogeneous set

Fig. 3: **Results achieved by the seven versions of the complex MA algorithm.** Simultaneous confidence intervals for all-pairwise comparisons of ranks between all versions applied to homogeneous and heterogeneous instances.

Table 3. Configuration selected by off-line tuning for $\mathcal{MM}$AS on the two sets of instances tackled, for each $\mathcal{MAX}$–$\mathcal{MIN}$ ant system algorithm and for each CPU time considered as stopping criterion [s].

	heterogeneous			homogeneous		
time	10	31	100	10	31	100
simple	1.0	1.0	1.5	1.0	1.0	1.5
complex	1.0	1.0	1.0	1.0	1.0	1.0

Table 4. Algorithm quality. Percentage of error obtained after 100 seconds by the literature version of the two versions of $\mathcal{MAX}$–$\mathcal{MIN}$ ant system, with respect to the best known solution of each instance.

	homogeneous	heterogeneous
simple	7.08287 %	10.84968 %
complex	0.29877 %	0.09805 %

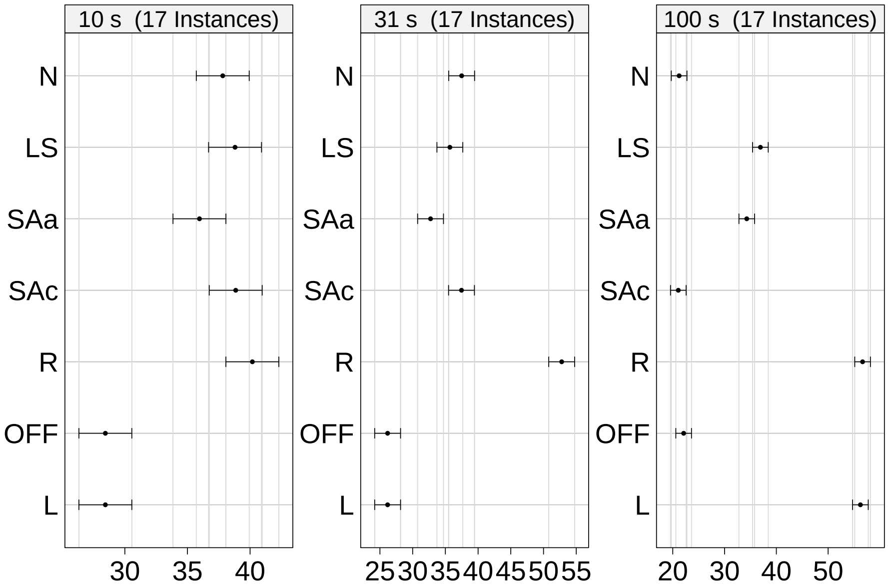
search. We implemented the on-line tuning methods described by Pellegrini et al. [8], and we tuned parameter α, that is, the exponent value used for the pheromone trails in the state transition rule.

Table 3 reports the configuration selected by off-line tuning for the two versions of $\mathcal{MM}$AS for different run-lengths. Table 4 shows the percentage error obtained by the literature version of the three algorithms in one run of 100 seconds on the homogeneous and heterogeneous instances excluded by the tuning phase, with respect to the best known solution of each instance. The best algorithm is the complex one, and the simple one is the worst performing. For each instance set the difference between all pairs of algorithms is statistically significant at the 95% confidence level, according to the Wilcoxon rank-sum test.

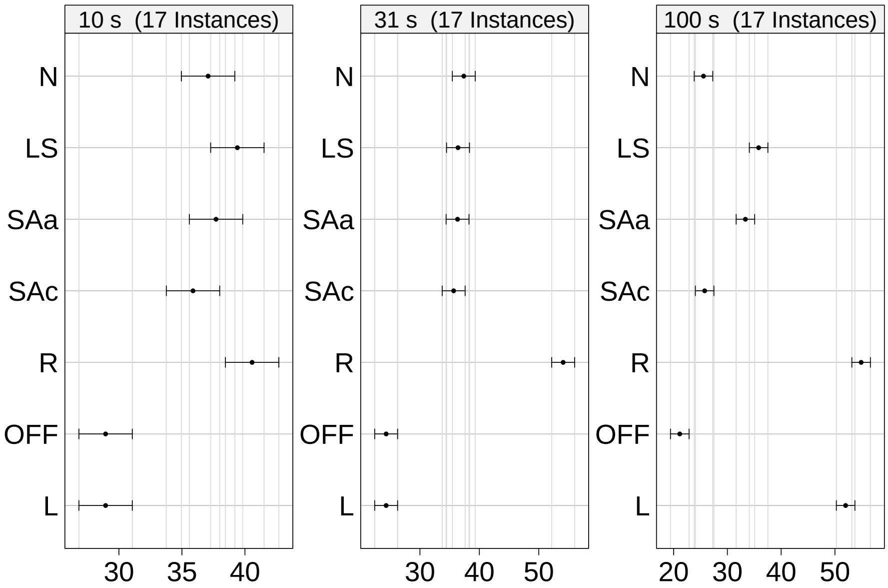
The results are even more strongly in favor of off-line tuning, even if some trend suggesting the existence of a relation between the relative performance of the tuning methods and the quality of the algorithms may be detected.

References

1. Francesca, G., Pellegrini, P., Stützle, T., Birattari, M.: Off-line and on-line tuning: A study on operator selection for a memetic algorithm applied to the qap (2010)
2. Merz, P., Freisleben, B.: Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. IEEE Transactions on Evolutionary Computation **4**(4) (2000) 337–352
3. Stützle, T., Hoos, H.H.: MAX–MIN ant system. Future Generation Computer Systems **16**(8) (2000) 889–914
4. Pellegrini, P., Stützle, T., Birattari, M.: "tuning MAX-MIN ant system with off-line and on-line methods". Technical Report TR/IRIDIA/2010-024, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium (2010)
5. Merz, P., Freisleben, B.: A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem. In: Proc. Congress on Evolutionary Computation, IEEE, Press (1999) 2063–2070

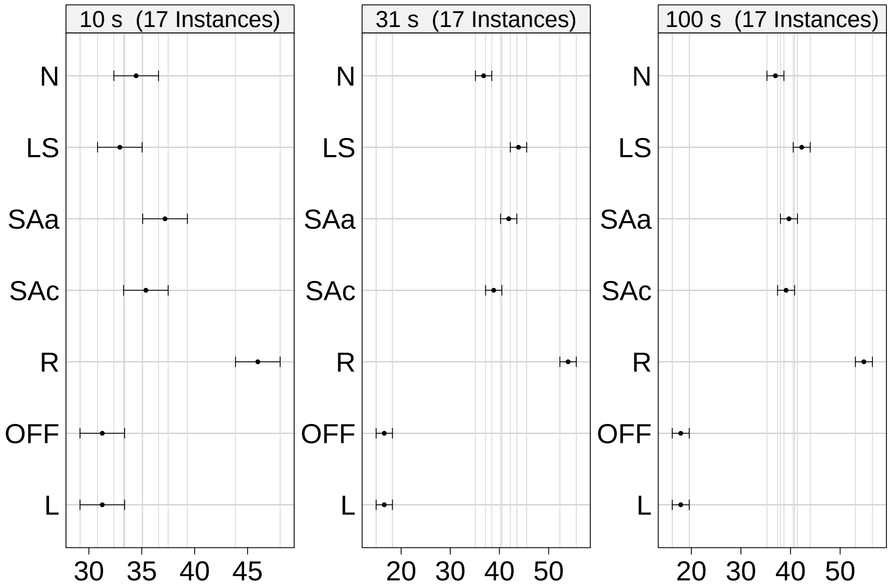


(a) homogeneous set

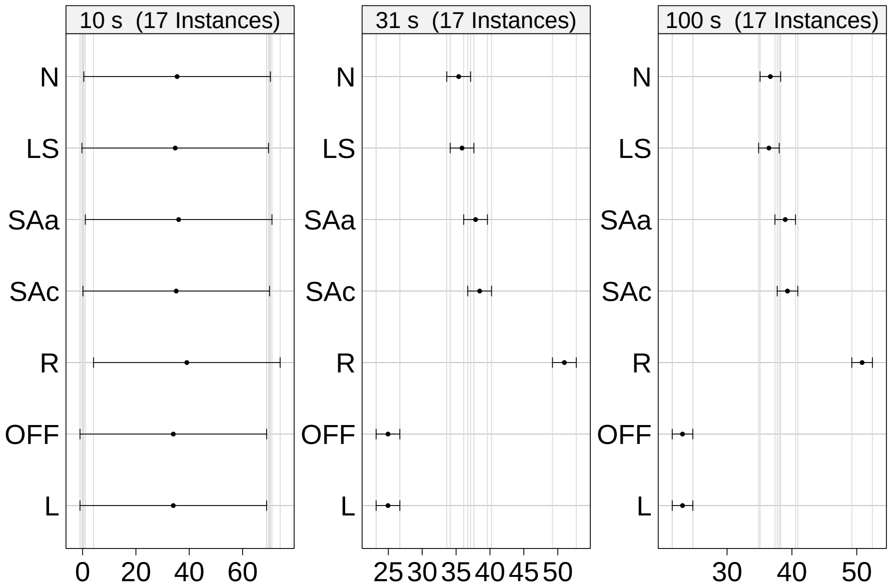


(b) heterogeneous set

Fig. 4: **Results achieved by the seven versions of the simple $\mathcal{MM}$AS algorithm.** Simultaneous confidence intervals for all-pairwise comparisons of ranks between all versions applied to homogeneous and heterogeneous instances.



(a) homogeneous set



(b) heterogeneous set

Fig. 5: **Results achieved by the seven versions of the complex $\mathcal{MM}$AS algorithm.** Simultaneous confidence intervals for all-pairwise comparisons of ranks between all versions applied to homogeneous and heterogeneous instances.

6. Burkard, R., Karisch, S., Rendl, F.: QAPLIB – A quadratic assignment problem library. Journal of Global Optimization (10) (1997) 391–403
7. Stützle, T., Fernandes, S.: New benchmark instances for the QAP and the experimental analysis of algorithms. In Gottlieb, J., Raidl, G., eds.: EvoCOP 2004. Volume 3004 of LNCS., Springer, Heidelberg, Germany (2004) 199–209
8. Pellegrini, P., Stützle, T., Birattari, M.: Off-line and on-line tuning: a study on MAX-MIN ant system for TSP. In Dorigo, M., et al., eds.: ANTS 2010. Volume 6234 of LNCS., Springer, Heidelberg, Germany (2010) 239–250