# A Detailed Analysis of the Population-Based Ant Colony Optimization Algorithm for the TSP and the QAP


Sabrina Oliveira
IRIDIA, CoDE, ULB
Brussels, Belgium
sabrina.oliveira@ulb.ac.be

Mohamed Saifullah Hussin
IRIDIA, CoDE, ULB
Brussels, Belgium
mbinhuss@ulb.ac.be

Thomas Stützle
IRIDIA, CoDE, ULB
Brussels, Belgium
stuetzle@ulb.ac.be

Andrea Roli
DEIS, Universitá di Bologna
Cesena, Italy
andrea.roli@unibo.it

Marco Dorigo
IRIDIA, CoDE, ULB
Brussels, Belgium
mdorigo@ulb.ac.be



## ABSTRACT

The population-based ant colony optimization algorithm (P-ACO) uses a very different pheromone update when compared to other ACO algorithms. In this work, we study P-ACO's behavior for solving the traveling salesman problem (TSP) and the quadratic assignment problem (QAP). In particular, we investigate the impact of a local search on P-ACO parameters and performance. The results clearly show that P-ACO is a very competitive tool whose parameters and behavior depend strongly on the problem tackled and on whether a local search is used.



## Keywords

Traveling salesman problem, quadratic assignment problem, population-based ant colony optimization


## Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; G.1.6 [**Numerical Analysis**]: Optimization

## General Terms

Algorithms

## 1. INTRODUCTION

In ACO, the way pheromone update is implemented differs across variants, and the choice of an appropriate pheromone update mechanism is essential to obtain effective ACO algorithms [1].

The population-based ACO (P-ACO) algorithm introduces a new memory and pheromone update [2, 3] but keeps the same solution construction as in most ACO algorithms. The pheromone update operates on two data structures: the *pheromone matrix* and the *solution archive*. P-ACO starts with an empty solution archive, $P$, and all the entries of the pheromone matrix are initialized to $\tau_0$, which has the same effect as the minimum pheromone trail limit in $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System (MMAS) [1]. At each iteration, a solution $\pi$ is added to the archive, until the archive size, $K$, is reached.

The general form of the P-ACO pheromone matrix is:

$$\tau_{ij} = \tau_0 + \Delta \sum_{k=1}^{|P|} w_k \cdot I_{ij}^k, \quad where \quad \Delta = \frac{\tau_{max} - \tau_0}{K}. \quad (1)$$

$\tau_{max}$ is a parameter and $I_{ij}^k$ is an indicator function that is equal to one if the solution component $ij$ is present in ant $k$ and zero otherwise. Every time a solution enters (or exits) $P$, the amount $\Delta \cdot w_k$ (or $-\Delta \cdot w_k$) is added to the pheromone matrix. A new solution replaces another one in the population matrix when the solution archive reaches $|P|$=$K$ solutions. As a result, P-ACO's pheromone update mechanism is faster than in other ACO algorithms.

Our objective is to study P-ACO's behavior on the TSP and QAP, as it has mainly been studied by a few other authors only [4], despite its potential.

## 2. EXPERIMENTS

To meet out objectives, experiments were performed using instances of the TSP and the QAP, ranging for both problems from small to relatively large instances. In our experiments, we studied the influence of P-ACO's main parameters on its performance and the interaction with the usage of local search. We compared it with MMAS, which was run with standard settings as suggested in [1]. In the following, we give a snapshot of our main results. For more details, we refer to the full version of this paper [4]. More detailed information and a full set of the plots is also available at `http://iridia.ulb.ac.be/supp/IridiaSupp2011-010`.

**Speed of pheromone update**. We measure the time spent by P-ACO and other algorithms for pheromone update and solution construction. P-ACO, MMAS and Ant Colony System (ACS) were considered for the TSP; for the QAP, ACS was not included. P-ACO is much faster than MMAS for the pheromone update for the TSP. On large instances, up to 90% of MMAS's computation time is used for pheromone update, while for P-ACO, this is a small percentage, allowing it to generate many more solutions than MMAS. With local search, the advantage of the fast P-ACO's phero-



*GECCO'11,* July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

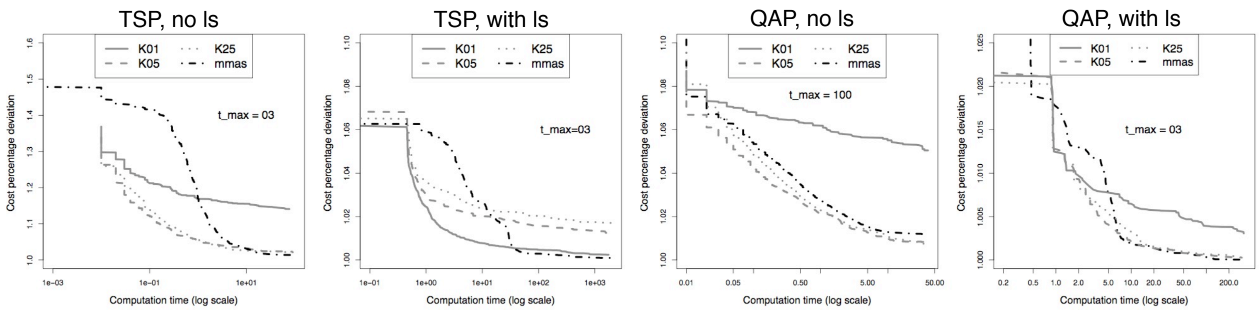


**Figure 1: Solution quality over time of MMAS and P-ACO using different values of $K$ across 10 trials for the best setting of $\tau_{max}$, without local search on instance d198 for TSP and on instance wil50 for QAP, with local search on instance u1817 for TSP, and on instance tai100b for QAP.**

mone update is reduced. The same happens if the computation of solution quality is computationally more demanding, as it is the case for the QAP; for QAP with local search the time spent for the pheromone update by P-ACO and MMAS is negligible and, hence, the speed advantage of P-ACO over MMAS becomes virtually irrelevant.

As a next step, we examined how P-ACO specific parameters influence the algorithm's behavior; in particular, we tested the algorithm with different settings of $K$ and $\tau_{max}$.

**Analysis of K**. The plots in Figure 1 show that, for both problems, without local search, P-ACO performs the worst when $K = 1$, and the best when $K = 25$, followed closely by $K = 5$, with negligible differences. If local search is applied, the best $K$ now becomes clearly one on the TSP; on the QAP, P-ACO with a larger value of $K$ performs better especially for larger instances. Note that in Figure 1, $\tau_{max}$ is 3 for the TSP while for the QAP it is also 3 when local search is applied but 100 when it is not applied. The choice of these parameter values is justified in the following.

**Analysis of $\tau_{\max}$**. In P-ACO, the ratio between $\tau_{max}$ and $\tau_{min}$, together with the setting of $K$, determines how strong the search intensification is. However, the changes $\tau_{max}$ shows a slitly different behavior when P-ACO is applied to different problems.

**Update of Solution Archive**. To compare the influence of the archive update strategy on algorithm performance, we tested three of the five strategies proposed in [2]: the age-based, quality-based, and elitist-based strategies. For the TSP, the strategies' performance is quite similar independently of whether local search is used. However, the quality-based strategy obtains better results. For all instances tested, P-ACO often outperforms MMAS, especially for a short computation time. For the QAP and without local search, the elitist-based strategy is almost as good as MMAS, while the other strategies show worse results. With local search, P-ACO performs better, where all strategies are competitive with MMAS.

**P-ACO with Restart**. P-ACO has a strong exploitation capability that allows a fast convergence to a good quality solution. However, its exploration during the search may be insufficient. As in MMAS, we restart by re-initializing the pheromone values to $\tau_0$ after $r$ iterations without improvement [4]. This mechanism improves P-ACO's performance, especially when local search is used.

A more extensive comparison showed that (i) for the TSP without local search, P-ACO performs better than MMAS on most instances, (ii) for the TSP with local search, the performance of MMAS and P-ACO are similar, (iii) for the QAP without local search, P-ACO performs better but (iv) with local search the opposite (better performance of MMAS over P-ACO) was observed. Overall, these results show that P-ACO with the restart procedure appears to be competitive with MMAS, regardless of whether local search is used.

# 3. CONCLUSIONS

In this paper, extensive experiments were conducted in order to discuss and analyze the P-ACO algorithm for the TSP and the QAP. New are the insights that the usage or not of a local search has strong impact on parameters settings for P-ACO applied to the TSP. In addition, we have shown that P-ACO shows early stagnation behavior and introduced a restart mechanism which has improved significantly the overall performance of P-ACO. We conclude that, with the restart procedure and the right configuration, P-ACO is competitive to the state-of-the-art ACO algorithms with the advantage of finding good solution quality in a shorter computation time. For more details we refer to [4].

**Acknowledgments**. This work was supported by the E-SWARM project, funded by an ERC Advanced Grant, and by the Meta-X project, funded by the Scientific Research Directorate of the French Community of Belgium. S. Oliveira is funded by a CAPES scholarship from the Brazilian government. T. Stützle and M. Dorigo acknowledge support from the Belgian F.R.S.-FNRS.

# 4. REFERENCES

[1] M. Dorigo and T. Stützle. *Ant Colony Optimization.* MIT Press, Cambridge, MA, 2004.

[2] M. Guntsch. Ant Algorithms in Stochastic and Multi-Criteria Environments. PhD thesis, Universität Fridericiana zu Karlsruhe, 2004.

[3] M. Guntsch and M. Middendorf. Applying population based ACO to dynamic optimization problems. In M. Dorigo et al., editors, *ANTS 2002*, volume 2463 of *LNCS*, pages 111–122. Springer, Heidelberg, 2002.

[4] S. Oliveira, M. S. Hussin, T. Stützle, A. Roli, and M. Dorigo. A Detailed Analysis of the Population-Based Ant Colony Optimization Algorithm for the TSP and the QAP. Technical Report TR/IRIDIA/2011-006, IRIDIA, ULB, 2011.