On $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System's parameters

Paola Pellegrini, Daniela Favaretto, Elena Moretti

Department of Applied Mathematics
University Ca' Foscari of Venice
Italy
paolap@pellegrini.it, favaret@unive.it, emoretti@unive.it

Abstract. The impact of the values of the most meaningful parameters on the behavior of $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System is analyzed. Namely, we take into account the number of ants, the evaporation rate of the pheromone, and the exponent values of the pheromone trail and of the heuristic measure in the random proportional rule. We propose an analytic approach to examining their impact on the speed of convergence of the algorithm. Some computational experiments are reported to show the practical relevance of the theoretical results.

Keywords: parameters, $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System.

1 Introduction

The assignment of values to the parameters of ACO algorithms is analyzed for the first time in [1]. In the following, a growing number of papers have been produced for finding the *optimal* values, or more in general for identifying the influence of the parameters on the behavior of the algorithms. These studies can be divided into two groups: the ones that propose a method for finding suitable parameter settings, and the ones that propose experimental analysis from which a sort of general trend can be deduced. Among others, we can locate in the first group the works by Botee and Bonabeau [2], Pilat and White [3], and Zaitar and Hiyassat [4] who use genetic algorithms for setting the parameters of ACO algorithms, and Randall [5] who uses an ACO algorithm itself. In the second group we can include Gaertner and Clark [6], who try to find a correlation between the structure of a problem instance and the optimal values of the parameters, and Socha [7] and Solnon [8], who propose computational studies concerning some parameters.

Another branch of the literature has considered the problem of tuning the parameters of metaheuristics, more in general. Among others, these include: Adenso-Díaz and Laguna [9] and Coy et al. [10] whose approaches are based on the *response surface methodology.* Bartz-Beielstein and Markon [11] propose a method to determine relevant parameter settings, based on statistical design of experiments, classical regression analysis, tree based regression and DACE (design and analysis of computer experiments) models. Birattari et al. [12] propose a procedure based on the Friedman two-way analysis of variance by ranks.

Finally, Battiti and Tecchioli [13] propose to tune the parameters while solving an instance, and Lau et al. [14] present a methodology called the *Visualizer for Metaheuristics Development Framework* (V-MDF).

Following this interest in the configurations of the parameters, the objective of this paper is to formalize the impact of the value chosen for the parameters of $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System [15, 16] on the speed of convergence to the best solution ants are able to find. In the following the term convergence will be used with this meaning.

Gaining understanding in this sense is important for two main reasons. First of all we want to stress the fact that it is not possible to define an *optimal* set of values for the parameters. While in general it is accepted that the values to assign depend from the problem and from the particular instances, it is not so infrequent to observe that some parameters are considered either *good* or *bad* (in terms of the average quality of the solution achieved) in absolute, without any reference to the computational time (t). The problem with this approach is that the *optimal* speed of convergence depends both on the instance and on the computational time available: If the solution is needed very fast, one might prefer a configuration of the parameters that reaches a local optimum, with respect to one that keeps exploring the search space and that probably in a longer time would reach a better local minimum. In this sense, analyzing the influence of each parameter on the searching behavior of the algorithm can be a way for emphasizing this element.

On the other hand, by gaining a deeper understanding of the dynamics underlying the algorithm, one might focus on a range of values of the parameters for the tuning phase. In this way, a finer choice would be possible.

For this analysis we consider a problem that can be represented on a graph $G = (N, A)$ with N set of nodes ($|N| = n$) and A set of arcs. For representing the time available we use an approximation: we suppose that the pheromone update is not time consuming. In other words, we consider the time in terms of number of solutions that can be built (T).

The paper is organized as follows. In Section 2 the relevant formulas characterizing $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System are presented. In Sections 3, 4 and 5 the parameters of the algorithm are studied. Finally in Section 6 some computational results are presented. The well known traveling salesman problem is considered as case study.

2 $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System

In $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System the pheromone update is performed after the activity of each colony of ants according to $\tau_{ij} = (1-\rho)\tau_{ij} + \Delta\tau_{ij}^b$, where $\Delta\tau_{ij}^b = 1/C_b$ if arc (i, j) belongs to the best solution b, and $\Delta\tau_{ij}^b = 0$ otherwise. C_b is the cost associated with solution b, and solution b is either the iteration-best solution or the best-so-far solution. Intuitively, if the iteration-best solution is used, the level of exploration is greater. The schedule according to which the solution to be exploited is chosen, is described by Dorigo and Stützle [17].

Another element characterizing ACO algorithms is the random-proportional rule. In particular, ant k being in node i and not having visited the nodes belonging to the set $N_k \subset N$, randomly chooses node $j \in N_k$ to move to. Each node $j \in N_k$ has a probability of being chosen described in the random proportional rule: $p_{ij} = [\tau_{ij}]^{\alpha} [\eta_{ij}]^{\beta} / (\sum_{h \in N_k} [\tau_{ih}]^{\alpha} [\eta_{ih}]^{\beta})$, where η_{ij} is a heuristic measure associated with arc (i, j) [17]. It is important to notice that this probability depends on the set of nodes not yet visited.

Finally, it is important to remember that the pheromone trail in $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System is bounded between τ_{MAX} and τ_{min}. Following [17], we use the following values: $\tau_{MAX} = 1/(\rho C_{best-so-far})$, and $\tau_{min} = [\tau_{MAX}(1 - \sqrt[n]{0.05})] / [(\frac{n}{2} - 1)\sqrt[n]{0.05}]$. At the beginning of a run, the best solution corresponds to the one found by the nearest neighbor heuristic (NN).

3 Number of ants m

Let us first of all analyze the effect of the number of ants on the behavior of the algorithm. One thing to notice is that, given a certain number of solutions T that can be built in the available run time, the number of ants m determines the number of iterations S that can be performed as $S = T/m$. A part from this element, the value of m affects the behavior of the algorithm for what is concerned the level of exploration. Given the pheromone update rule and the update schedule, the level of exploration mainly decided by the solutions used for the update. This is due to the fact that if only few solutions are used, after few iterations, only the arcs belonging to them will have a significant probability of being chosen. If the variation of the iteration best solution is small, then, the convergence will be fast. Let br_s be the iteration best solution at iteration s, and $BR = \{r_1, r_2, ..., r_{|BR|}\}$ the set of previous iteration best solutions. If a solution has been the iteration best more than once, obviously it will be inserted in BR only the first time. Let us analyze the probability of having as iteration best solution at iteration s a solution belonging to BR. Note that, given the solution construction procedure, we can easily associate to each feasible solution r a probability of being built ($\bar{p}_r$).

Let Ω be the set of all the possible solutions and $R_r = \{q \in \Omega : C_q \geq C_r\}$. The probability of having as iteration best solution at iteration s a solution r is: $p(br_s = r) = \bar{p}_r \left(\sum_{q \in R_r} \bar{p}_q\right)^{(m-1)}$. It is the product of the probability of having one ant constructing exactly r and all the other ants constructing solutions $q \in R_r$. In the following we consider all the solutions as having different costs, so that the ordering of the solutions is not ambiguous. Since $\sum_{q \in R_r} \bar{p}_q \leq 1$ (the case of equality is true only if r is the global optimum), $p(br_s = r)$ is decreasing in m. The meaning of this conclusion is that the higher the number of ants, the lower the probability of selecting as iteration best solution a specific one. This reasoning can be extended considering that at iteration s, $|BR|$ solutions $(r_i, i = 1, ..., |BR|)$ have already been selected. In particular, the probability of selecting as iteration best at iteration s a solution in BR is equal to:

$$p(br_s \in BR) = \sum_{r \in BR} \bar{p}_r \Big(\sum_{q \in R_r} \bar{p}_q \Big)^{(m-1)}. \tag{1}$$

This value is decreasing in m and increasing in $|BR|$. For the first property it is sufficient to observe that $\sum_{q \in R_r} \bar{p}_q < 1$ (we do not consider the global optimum). For the property related to $|BR|$, it is clear that the result of a sum of non negative terms is increasing in the number of addends, the probabilities $\bar{p}_r$ being equal. Obviously $|BR|$ is non decreasing in s.

The conclusion of this reasoning is that the higher the number of ants, the greater the exploration. On the other hand, the higher the number of iterations, the greater the exploitation of the cumulated knowledge. Moreover, given the available computational time, the greater the number of ants, the smaller the number of total iterations. It is clear, then, that there is a trade-off to be solved. Remark that this reasoning is independent from the probability of choice of any particular solution $\bar{p}$, if this probability is not null for all the solutions. This property is ensured by $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System through the imposition of a positive lower bound of the pheromone.

4 Evaporation rate ρ

The parameter ρ is present in the pheromone update rule. It fixes how much pheromone evaporates. The relevance of this parameter is related to the level of exploration of the search space performed: If ρ is high, the pheromone on the arcs belonging to solutions built a few iterations before will be roughly equal to the one on the arcs that have never been selected. In this way, the search will not be much biased toward the already visited areas.

On an arc (i, j) which has never been used, the pheromone at iteration $\bar{s}$ is equal to $(1-\rho)\tau_{ij} = (1-\rho)^{\bar{s}}\tau_0 = (1-\rho)^{\bar{s}}/(\rho C_{NN})$. Clearly this is a decreasing function of ρ. What we are interested in is the influence of the value of this parameter on the level of exploration. In particular, we want to know what are the conditions for having the minimum probability of choosing, after $\bar{s}$ iterations, an arc that has never been part of an iteration best solution, and is then supposed to be of *bad* quality. This objective is achieved by setting the pheromone in arc (i, j) equal to τ_{min}. As an approximation for this value we use τ_{min} at iteration 0, i.e. $[1/(\rho C_{NN})(1-\sqrt[n]{0.05})]/[(\frac{n}{2}-1)\sqrt[n]{0.05}]$. The investigation is then referred to the value of ρ such that $\tau_{ij} \leq \tau_{min}$:

$$(1-\rho)^{\bar{s}} \frac{1}{\rho C_{NN}} \leq \frac{\frac{1}{\rho C_{NN}}(1-\sqrt[n]{0.05})}{(\frac{n}{2}-1)\sqrt[n]{0.05}} \quad \Rightarrow \quad \rho \geq 1 - \sqrt[\bar{s}]{\frac{(1-\sqrt[n]{0.05})}{(\frac{n}{2}-1)\sqrt[n]{0.05}}}. \tag{2}$$

In this way, it is possible to fix a relation between ρ, the number of nodes of the graph (n) and the number of iterations ($\bar{s}$) after which an arc that has never been part of a solution used for the pheromone update has the minimum possible probability of being chosen.

Proposition 1. *If, given n, $\bar{s}$ is such that* $\rho \geq 1 - \sqrt[\bar{s}]{\frac{(1-\sqrt[n]{0.05})}{(\frac{n}{2}-1)\sqrt[n]{0.05}}}$, *then* $\rho \geq 1 - \sqrt[\bar{s}']{\frac{(1-\sqrt[n]{0.05})}{(\frac{n}{2}-1)\sqrt[n]{0.05}}}$, $\forall \bar{s}' \geq \bar{s}$.

Proof. $\rho \geq 1 - \sqrt[\bar{s}]{\frac{(1-\sqrt[n]{0.05})}{(\frac{n}{2}-1)\sqrt[n]{0.05}}}$, $0 < \rho < 1 \Rightarrow (1-\rho)^{\bar{s}} \geq (1-\rho)^{\bar{s}'}$, $\forall \bar{s}' \geq \bar{s} \Rightarrow$
$(1-\rho)^{\bar{s}'} \leq \frac{(1-\sqrt[n]{0.05})}{(\frac{n}{2}-1)\sqrt[n]{0.05}} \Rightarrow \rho \geq 1 - \sqrt[\bar{s}']{\frac{(1-\sqrt[n]{0.05})}{(\frac{n}{2}-1)\sqrt[n]{0.05}}}$, $\forall \bar{s}' \geq \bar{s}$. □

Proposition 2. *If, given $\bar{s}$, n is such that* $\rho \geq 1 - \sqrt[\bar{s}]{\frac{(1-\sqrt[n]{0.05})}{(\frac{n}{2}-1)\sqrt[n]{0.05}}}$, *then*[1] $\forall n'$ *such that* $3 \leq n' \leq n$, $\quad \rho \geq 1 - \sqrt[\bar{s}]{\frac{(1-\sqrt[n']{0.05})}{(\frac{n'}{2}-1)\sqrt[n']{0.05}}}$.

Proof. $\rho \geq 1 - \sqrt[\bar{s}]{\frac{(1-\sqrt[n]{0.05})}{(\frac{n}{2}-1)\sqrt[n]{0.05}}} \Rightarrow \sqrt[n]{0.05}\left[(1-\rho)^{\bar{s}}\left(\frac{n}{2}-1\right)+1\right] \leq 1$.
$\sqrt[n]{0.05}$ is an increasing function of n, then

$$\sqrt[n']{0.05}\left[(1-\rho)^{\bar{s}}\left(\frac{n}{2}-1\right)+1\right] \leq \sqrt[n]{0.05}\left[(1-\rho)^{\bar{s}}\left(\frac{n}{2}-1\right)+1\right]. \qquad (3)$$

Moreover, $\forall n' \leq n$

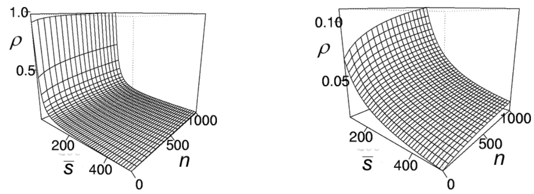
$$\sqrt[n']{0.05}\left[(1-\rho)^{\bar{s}}\left(\frac{n'}{2}-1\right)+1\right] \leq \sqrt[n']{0.05}\left[(1-\rho)^{\bar{s}}\left(\frac{n}{2}-1\right)+1\right], \qquad (4)$$

from which the thesis is verified. □

Given propositions 1 and 2, it is quite easy to fix a lower bound for the value of ρ, both in a quite general case and in a specific one. For the first observation, one can compute the value of ρ which allows the algorithm to neglect the *bad* arcs (in terms of the average quality of the solutions they belong to) after a small number of iterations when dealing with a very big instance. To this aim, let $\bar{s} = 100$ and $n = 1000$, which implies $\rho \sim 0.1$. If one sets $\rho = 0.1$, after $\bar{s}' > \bar{s}$ iterations, for sure the algorithm will have neglected the *bad* arcs. In the same way, if n decreases, $\rho = 0.1$ will imply that after $\bar{s}$ iterations, the algorithm will have neglected the *bad* arcs.

In addition to this general purpose observation, if one needs to tackle instances of equal (or similar) size, one can fix a meaningful value for ρ after estimating $\bar{s}$. Clearly this estimate will depend on the available computational time. Figure 1 represents the trend followed by the value of this parameter when $\bar{s}$ and n vary. It is easy to see that the number of iterations is the leading force, at least until a certain threshold. Nonetheless, the number of nodes has a remarkable impact as well.

[1] If $n < 3$ the value of τ_{min} is not defined.



(a) $3 \leq \bar{s} \leq 500$ (b) $100 \leq \bar{s} \leq 500$

Fig. 1. Value of ρ necessary for having $\tau_{ij} = \tau_{min}$ on a never reinforced arc (i, j).

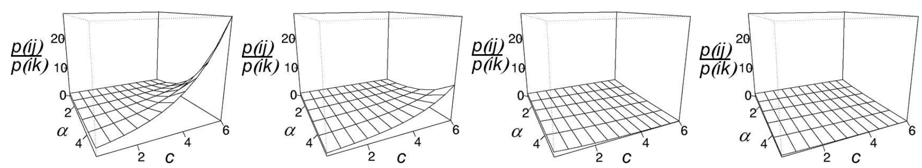
5 Exponent values α and β

The last parameters we are going to consider for $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System are α and β. They represent the exponent of the pheromone level and the heuristic measure in the random proportional rule, respectively. Their main role consists in emphasizing the differences between arcs.

Instead of studying the trend of the probability of choosing the single arc, we consider the ratio between the probability of choosing two arcs (i, j) and (i, k). By analyzing this element it is possible not to consider the set of nodes still to visit. Let us write β as $c\alpha$, with $c \geq 0$. The ratio we want to study, then, is reported in formula (5).

$$\frac{p_{ij}}{p_{ik}} = \frac{[\tau_{ij}]^{\alpha} [\eta_{ij}]^{c\alpha}}{[\tau_{ik}]^{\alpha} [\eta_{ik}]^{c\alpha}} = \left[\frac{\tau_{ij}}{\tau_{ik}}\right]^{\alpha} \left[\frac{\eta_{ij}}{\eta_{ik}}\right]^{c\alpha} = \left[\frac{\tau_{ij}}{\tau_{ik}} \left(\frac{\eta_{ij}}{\eta_{ik}}\right)^{c}\right]^{\alpha} = f(\alpha, c). \tag{5}$$

Remark that being the pheromone limited by a positive lower bound, and being the length of the arcs a finite number, $\tau_{(\cdot)}$ and $\eta_{(\cdot)}$ are always strictly positive. Then, the sign of the first partial derivative with respect to α depends on $\ln\left[(\tau_{ij}/\tau_{ik})(\eta_{ij}/\eta_{ik})^{c}\right]$. This quantity is positive if and only if $(\tau_{ij}/\tau_{ik})(\eta_{ij}/\eta_{ik})^{c} > 1$. On the other hand, the sign of the first partial derivative with respect to c



(a) $\frac{\tau_{ij}}{\tau_{ik}} > 1, \frac{\eta_{ij}}{\eta_{ik}} > 1$ (b) $\frac{\tau_{ij}}{\tau_{ik}} < 1, \frac{\eta_{ij}}{\eta_{ik}} > 1$ (c) $\frac{\tau_{ij}}{\tau_{ik}} > 1, \frac{\eta_{ij}}{\eta_{ik}} < 1$ (d) $\frac{\tau_{ij}}{\tau_{ik}} < 1, \frac{\eta_{ij}}{\eta_{ik}} < 1$

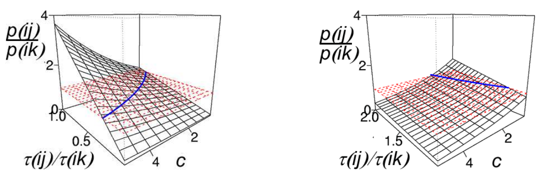
Fig. 2. Ratio of the probabilities related to the choice to arc (i, j) and arc (i, k).

depends on $\ln[\eta_{ij}/\eta_{ik}]$, which is positive if and only if $\eta_{ij}/\eta_{ik} > 1$. A graphical representation of its trend is shown in Figure 2. The value of c determines both the magnitude of the variation, and the increase or decrease of the function. Then, let us have a look at function $g(c) = (\tau_{ij}/\tau_{ik})\,(\eta_{ij}/\eta_{ik})^c$. In particular we are interested in knowing in which cases the function is greater than 1.

$$g(c) > 1 \Rightarrow \ln \frac{\tau_{ij}}{\tau_{ik}} + c \ln \frac{\eta_{ij}}{\eta_{ik}} > 0 \Rightarrow c \begin{cases} > -\frac{\ln \tau_{ij}/\tau_{ik}}{\ln \eta_{ij}/\eta_{ik}} \text{ if } \eta_{ij} > \eta_{ik} \\ < -\frac{\ln \tau_{ij}/\tau_{ik}}{\ln \eta_{ij}/\eta_{ik}} \text{ if } \eta_{ij} < \eta_{ik} \end{cases} \tag{6}$$

Following the literature we consider only $\alpha, \beta \geq 0$. Let us first of all analyze the first inequality of (6). If $\tau_{ij} > \tau_{ik}$, then $\ln[\tau_{ij}/\tau_{ik}] > 0$ and the whole quantity on the right hand side of the inequality is negative, so there is no restriction on c for having $g(c)$ positive. If $\tau_{ij} < \tau_{ik}$, instead, there is a meaningful lower bound for c. A similar and opposite reasoning holds for the second inequality of (6): if $\tau_{ij} < \tau_{ik}$, then $\ln[\tau_{ij}/\tau_{ik}] < 0$ and the whole quantity on the right hand side of the inequality is negative, so there is no possible value of c such that $g(c)$ is positive. If $\tau_{ij} > \tau_{ik}$, instead, there is a meaningful upper bound for c.

Clearly the ratios between $\tau_{(\cdot)}$'s and between $\eta_{(\cdot)}$'s depend on the arcs we choose as (i, j) and (i, k). Moreover, as for what τ_{ij}/τ_{ik} is concerned, it depends on the behavior of the algorithm. Figure 3 represents the variation of $g(c)$ as a function of τ and η. In particular, we keep constant through the graphics the value of η_{ij}/η_{ik} and we vary the ratio between the pheromone levels. This schema follows the behavior of ACO algorithms in cases the heuristic measure is static. The values selected are $\eta_{ij}/\eta_{ik} = 1.1$ for Figure 3(a) and $\eta_{ij}/\eta_{ik} = 0.9$ for Figure 3(b). As it can be seen, there is an interval in which, even if the position with respect to 1 of τ_{ij}/τ_{ik} and $\eta_{ij}\eta_{ik}$ are opposite, for a while $g(c)$ keeps on following the sign of $1 - (\eta_{ij}/\eta_{ik})$. The greater c, the wider this interval. This observation is robust with respect to the value of η_{ij}/η_{ik}.



(a) $\frac{\eta_{ij}}{\eta_{ik}} > 1$ (b) $\frac{\eta_{ij}}{\eta_{ik}} < 1$

Fig. 3. Ratio of the probabilities related to the choice of arc (i, j) and arc (i, k).

To sum up the reasoning on α and β, α amplifies the differences between the *good* and the *bad* arcs. The value of $c = \beta/\alpha$, instead, tells us how to distinguish the *good* from the *bad* arcs in case the heuristic information and the pheromone values lead to discordant orders. In particular, the higher the value of c, the more the order is driven by the heuristic information.

6 Experiments

The experimental analysis proposed is based on the traveling salesman problem (TSP). We consider the ACOTSP program implemented by Thomas Stützle as a companion software for [17]. The code has been released in the public domain and is available for free download on www.aco-metaheuristic.org/aco-code/. The TSP has been object of many studies, both practical and theoretical (see for example [18–20]). We consider this problem as a case study.

The experiments proposed aim at showing that the implications of the previous sections are clearly detectable in practise. In this sense, we need a method for identifying *good* combinations of values of the parameters when the computational time available changes. We will read the configurations selected in terms of the speed of convergence they imply.

We chose the F-Race procedure [21, 12] for selecting the values of the parameters. F-Race is a racing algorithm for choosing a combination of values (a candidate configuration) from a predefined range. A racing algorithm consists in generating a sequence of nested sets of candidate configurations to be considered at each step. The set characterizing a specific step is obtained by possibly discarding some configurations that appear to be suboptimal on the basis of the information available. This cumulated knowledge is represented by the behavior of the algorithm for which the tuning is performed, when using different candidates configurations. For each instance (each representing one step of the race) the ranking of the results obtained using the different configurations is computed and a statistical test is performed for deciding whether to discard some candidates from the following experiments. F-Race is based on the Friedman two-way analysis of variance by ranks [22].

The range of values considered for each parameter is the one that in our eyes one would test after the analysis of the literature. In particular the candidate configurations are 192. They are all those obtainable from combining the following values: $m \in \{50, 100, 200, 300\}$, $\rho \in \{0.02, 0.04, 0.06, 0.08\}$, $\alpha \in \{1, 2, 3\}$, $\beta \in \{2, 3, 4, 5\}$. Two sets of 220 instances are used. In one set each instance includes 300 customers. In the other one 600 customers are considered. The instances are generated through `portgen`, the instance generator adopted in the DIMACS TSP Challenge. In particular, the ones we consider here consist of two dimensional integer-coordinate cities grouped in clusters that are uniformly distributed in a square of size $10^6 \times 10^6$. They are available on the web page www.paola.pellegrini.it. On each set of instances, the F-Race is applied six times, varying the computational time available t in the set $\{5, 10, 30, 60, 90, 120\}$ seconds. The experiments are run on a processor AMD Athlon 1000 Mhz, 772 MB of memory, running GNU/Linux 2.4.20. No local search is applied, due to the fact that we want to investigate the relation between the values of the parameters and the speed of convergence, and we are not interested in the absolute quality of the solution. The candidate configuration chosen by F-Race for each set of instances/computational time are reported in Table 1. Beside the values of the parameters selected, the table reports the approximate number of tours that can be built in the available time (T), the total number of iterations performed

(S), and the value of $c = \beta/\alpha$. The heuristic measure we consider is the typical one used for the TSP, i.e. the inverse of the length of arcs.

Table 1. Configurations chosen by F-race with different computational time available.

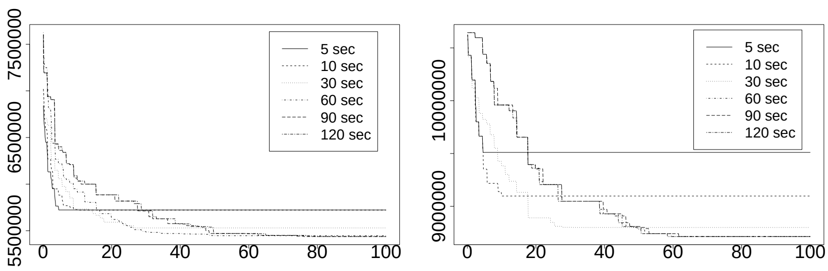
n	t	m	ρ	β	α	$\Rightarrow$	T	$S = T/m$	c
300	5	100	0.08	5	2		7000	70	2.5
300	10	100	0.08	4	2		14000	140	2
300	30	100	0.08	4	1		42000	420	4
300	60	100	0.08	3	1		84000	840	3
300	90	200	0.08	3	1		126000	630	3
300	120	200	0.08	3	1		168000	840	3
600	5	50	0.08	5	3		1700	34	1.66
600	10	50	0.08	5	3		3400	68	1.66
600	30	100	0.08	5	2		10200	102	2.5
600	60	200	0.08	4	2		20400	102	2
600	90	200	0.08	4	2		30600	153	2
600	120	200	0.08	4	2		40800	204	2

The trend followed by the values of the parameters are clear. They respect the expectations coming from the previous analysis. In particular it can be observed that the value of m increases with the increase of the computational time available. According to Section 3 this can be read as an increase of the level of exploration of the search space. The values of m are quite different through the cardinality of the set of nodes n. This is due to the fact that given the time available, the number of tours that can be constructed is noticeably different. Moreover, when considering different computational times which lead to the construction of a similar number of solutions, one can observe that the number of ants increases with n (see for example $n = 300, T = 42000 \Rightarrow m = 100$ and $n = 600, T = 20400 \Rightarrow m = 200$). This can be read as a greater need of exploration in case of a greater number of nodes. The explanation for this phenomenon can be found in the fact that the greater the number of nodes, in general the more complex the search space, and so the greater the risk of being entrapped in a local minimum.

The trend followed by the values of α mimics the prediction of Section 5. In fact, it is decreasing with the time available, reflecting the postposition of the need of convergence. For what concerns c, we can observe that, α being equal, its value is inversely correlated with the number of solutions that can be constructed. When α varies, the value of c changes in the opposite direction. When the value of α is high, the convergence is fast even if we do not consider the value of c. As a consequence, keeping c quite low is a way for smoothing the trend. On the other hand, speeding up the convergence is not the only role of c. As discussed in Section 5, it implies whether it is the heuristic information or the pheromone trail to state the distinction between the *good* from the *bad* arcs in case the respective indications are discordant. The higher the value of c, the more the decision is driven by the heuristic information. When we consider this element with the negative correlation between c and T, we can deduce that the earlier the algorithm needs to converge, the more it has to give importance to the heuristic measure, which has a more immediate link with the instance than the pheromone trail. In a similar way, let us consider the relation of c with the value of α. If the latter implies a very fast convergence, in general the algorithm will be reluctant to accept the indications of previous ants (and so of the pheromone trail) in case they are in contrast with the ones of the heuristic information. This is due to the fact that if they are misleading it will not be able to neglect them very soon.

Finally, it is not possible here to observe the trend followed by the value of ρ. The value selected, in fact, is always the same ($\rho = 0.08$), and it corresponds to the maximum value in the range set. Nonetheless, the choice has been motivated by the analysis of the literature and in particular of Dorigo and Stützle [17], where a value of 0.02 is proposed. By considering formula (2) it is possible to compute the number of iterations after which the pheromone on *bad* arcs becomes equal to τ_{min} with $\rho = 0.08$ and instances of 300 and 600 nodes. This quantity is equal to 116 and 132, respectively. These values are probably quite good when the total number of iterations is higher than 400-500, while it may be too low for shorter runs.

Figure 4 represents the trends followed by the values of the best solutions (C_b) found by $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System with the configurations of parameters reported in Table 1 as function of the computational time (t). As it can be seen, the configurations selected by F-Race are the best performing up to the time available for the respective runs.



(a) 300 nodes.

(b) 600 nodes.

Fig. 4. Value of (C_b) found by the different configurations depending on t.

7 Conclusion

The relevance of the values of the parameters when dealing with metaheuristics is recognized in the literature. In this paper we analyze $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System: Theoretical aspects of the impact of the values of the parameters on its behavior are investigated. Some relations between the values chosen and the speed of convergence of the algorithm are proposed. Computational experiments are reported to show the practical reflections of the theoretical results.

Once fixed the constraints that one must satisfy, such as the characteristics of the instances to tackle and the computational time available, the comprehension of the impact of the parameters can give some indications about the range to use for the tuning phase. Further possible developments of this study can be the analysis of different problems and other ACO algorithms.

References

1. Colorni, A., Dorigo, M., Maniezzo, V.: An investigation of some properties of an "ant algorithm". In Männer, R., Manderick, B., eds.: PPSN, Brussels, Belgium, Elsevier (1992) 515–526
2. Botee, H., Bonabeau, E.: Evolving ant colony optimization. Advanced Complex Systems **1** (1985) 149–159
3. Pilat, M.L., White, T.: Using genetic algorithms to optimize acs-tsp. In Dorigo, M., Caro, G.D., Sampels, M., eds.: ANTS '02: Proceedings of the Third International Workshop on Ant Algorithms, London, UK, Springer-Verlag (2002) 282–287
4. Zaitar, R., Hiyassat, H.: Optimizing the ant colony optimization using standard genetic algorithm. In Hamza, M., ed.: Artificial Intelligence and Applications, Innsbruck, Austria, IASTED/ACTA Press (2005) 130–133
5. Randall, M.: Near parameter free ant colony optimisation. In Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T., eds.: ANTS Workshop. Volume 3172 of Lecture Notes in Computer Science., Brussels, Belgium, Springer (2004) 374–381
6. Gaertner, D., Clark, K.: On optimal parameters for ant colony optimization algorithms. In Arabnia, H., Joshua, R., eds.: IC-AI 2005, Las Vegas, USA (2005) 83–89
7. Socha, K.: The influence of run-time limits on choosing ant system parameters. In Cantu-Paz, E., Livermore, L., Balakrishnan, K., Banzhaf, W., Bentley, P., Dasgupta, L.C.D., Jong, K.D., nad F. Herrera, J.F., Langdon, W., Lutton, E., Mazumder, P., Michielssen, E., Pedrycz, W., Roy, R., Rudnick, E., Soule, M.S.T., Spector, L., Verdegay, J., eds.: Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2003. Volume 2611 of LNCS., Chicago, USA, Springer-Verlag (2003) 49–60
8. Solnon, C.: The influence of run-time limits on choosing ant system parameters. In Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G., eds.: EvoWorkshop 2002. Volume 2279 of LNCS., Kinsale, Ireland, Springer-Verlag (2002) 163–172
9. Adenso-Díaz, B., Laguna, M.: Fine-tuning of algorithms using fractional experimental designs and local search. (to appear in *Operations Research*)
10. Coy, S., Golden, B., Runger, G., Wasil, E.: Using experimental design to find effective parameter settings for heuristics. Journal of Heuristics **7**(1) (2001) 77–97

11. Bartz-Beielstein, T., Markon, S.: Tuning search algorithms for real-world applications: A regression tree based approach. In Greenwood, G., ed.: Proc. 2004 Congress on Evolutionary Computation (CEC'04), Piscataway NJ, IEEE Press (2004) 1111–1118
12. Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K.: A racing algorithm for configuring metaheuristics. In Langdon, W., ed.: GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, CA, USA, Morgan Kaufmann Publishers (2002) 11–18
13. Battiti, R., Tecchioli, G.: The reactive tabu search. ORSA Journal on Computing **6** (1994) 126–585
14. Lau, H., Wan, W., Halim, S.: Tuning tabu search strategies via visual diagnosis. In Doerner, K., Gendreau, M., Greistorfer, P., Gutjahr, W., Hartl, R., Reimann, M., eds.: Proceedings of Metaheuristics International Conference (MIC 2005), Vienna, Austria (2005) 630–636
15. Stützle, T., Hoos, H.: Improvements on the ant system: introducing the max-min ant system. In Albrecht, R., Smith, G., Steele, N., eds.: Proceedings of Artificial Neural Nets and Genetic Algorithms 1997, Norwich, U.K., Springer-Verlag (1998) 245–249
16. Stützle, T., Hoos, H.: Max-min ant system. Future Generation Computer Systems **16**(8) (2000) 889–914
17. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
18. Bullnheimer, B., Hartl, R.F., Strauss, C.: A new rank based version of the ant system: A computational study. Central European Journal for Operations Research and Economics **7**(1) (1999) 25–38
19. Colorni, A., Dorigo, M., Maffioli, F., Maniezzo, V., Righini, G., Trubian, M.: Heuristics from nature for hard combinatorial problems. International Transactions in Operational Research **3**(1) (1996) 1–21
20. Stützle, T., Hoos, H.: The max-min ant system and local search for the traveling salesman problem. In Angeline, P., ed.: Proceedings of the IEEE International Conference on Evolutionary Computation, Indianapolis, USA, Springer-Verlag (1997) 308–313
21. Birattari, M.: The problem of tuning metaheuristics as seen from a machine learning perspective. PhD thesis, Université Libre de Bruxelles, Brussels, Belgium (2005)
22. Friedman, J.: Multivariate adaptive regression splines. The Annals of Statistics **19** (1991) 1–141