Chapter 10

THE ACO/F-RACE ALGORITHM FOR COMBINATORIAL OPTIMIZATION UNDER UNCERTAINTY

Mauro Birattari, Prasanna Balaprakash, and Marco Dorigo
IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
{mbiro,pbalapra,mdorigo}@ulb.ac.be

Abstract The paper introduces *ACO/F-Race*, an algorithm for tackling combinatorial optimization problems under uncertainty. The algorithm is based on *ant colony optimization* and on *F-Race*. The latter is a general method for the comparison of a number of candidates and for the selection of the best one according to a given criterion. Some experimental results on the PROBABILISTIC TRAVELING SALESMAN PROBLEM are presented.

Keywords: Ant colony optimization, combinatorial optimization under uncertainty

1. Introduction

In a large number of real-world combinatorial optimization problems, the objective function is affected by uncertainty. In order to tackle these problems, it is customary to resort to a probabilistic model of the value of each feasible solution. In other words, a setting is considered in which the cost of each solution is a *random variable*, and the goal is to find the solution that minimizes some *statistics* of the latter. For a number of practical and theoretical reasons, it is customary to optimize with respect to the *expectation*. This reflects a risk-neutral attitude of the decision maker. Theoretically, for a given probabilistic model, the expectation can always be computed but this typically involves particularly complex analytical manipulations and computationally expensive procedures. Two alternatives have been discussed in the literature: *analytical approximation* and *empirical estimation*. While the former explicitly relies on the underlying probabilistic model for approximating the expectation, the latter estimates the expectation through *sampling* or *simulation.*

In this paper we introduce *ACO/F-Race*, an *ant colony optimization* algorithm [8] for tackling combinatorial optimization problems under uncertainty with the *empirical estimation* approach. *F-Race* [6, 5] is an algorithm for the comparison of a number of candidates and for the selection of the best one. It has been specially developed for tuning metaheuristics.[1] In the present paper, *F-Race* is used in an original way as a component of an *ant colony optimization* algorithm. More precisely, it is adopted for selecting the *best-so-far* ant, that is, the ant that is appointed for updating the pheromone matrix.

The main advantage of the *estimation* approach over the one based on *approximation* is generality: Indeed, a sample estimate of the expected cost of a given solution can be simply obtained by averaging a number of realizations of the cost itself. Conversely, computing a profitable approximation is a problem-specific issue and requires a deep understanding of the underlying probabilistic model. Since *ACO/F-Race* is based on the *empirical estimation* approach, it is straightforward to apply it to a large class of combinatorial optimization problems under uncertainty. For definiteness, in this paper we consider an application of *ACO/F-Race* to the PROBABILISTIC TRAVELING SALESMAN PROBLEM, more precisely to its *homogeneous* variant [11]. An instance of the PROBABILISTIC TRAVELING SALESMAN PROBLEM (PTSP) is defined as an instance of the well known TRAVELING SALESMAN PROBLEM (TSP), with the difference that in PTSP each city has a given probability of requiring being visited. In this paper we consider the *homogeneous* variant, in which the probability that a city must be visited is the same for all cities. PTSP is here tackled in the *a priori* optimization sense [1]: The goal is to find an *a priori* tour visiting all the cities, which minimizes the expected length of the associated *a posteriori* tour. The *a priori* tour must be found prior to knowing which cities indeed require being visited. The associated *a posteriori* tour is computed *after* knowing which cities need being visited, and is obtained by visiting them in the order in which they appear in the *a priori* tour. The cities that do not require being visited are simply skipped. This problem was selected as the first problem for testing the *ACO/F-Race* algorithm for two main reasons: First, PTSP is particularly simple to describe and to handle. In particular, the *homogeneous* variant is rather convenient since a single parameter,

[1] A public domain implementation of *F-Race* for R is available for download [4]. R is a language and environment for statistical computing that is freely available under the GNU GPL license.

that is, the probability that each city requires being visited, defines the "stochastic character" of an instance: When the probability is one, we fall into the deterministic case; as it decreases, the normalized standard deviation of the cost of a given solution increases steadily. We can informally conclude that an instance of the *homogeneous* PTSP becomes *more and more stochastic* as the probability that cities require being visited decreases. This feature is particularly convenient in the analysis and visualization of experimental results. Second, some variants of *ant colony optimization* have been already applied to PTSP: Bianchi *et al.* [3, 2] proposed *pACS*, a variant of *ant colony system* in which an *approximation* of the expected length of the *a posteriori* tour is optimized; Gutjahr [9, 10] proposed *S-ACO*, in which an *estimation* of the expected length of the *a posteriori* tour is optimized. *ACO/F-Race* is similar to *S-ACO*. The main difference lies in the way solutions are compared and selected.

The rest of the paper is organized as follows: Section 2 discusses the problem of estimating, on the basis of a sample, the cost of a solution in a combinatorial optimization problem under uncertainty. Section 3 introduces the *ACO/F-Race* algorithm. Section 4 reports some results obtained by *ACO/F-Race* on PTSP. Section 5 concludes the paper and highlights future research directions.

2. The empirical estimation of stochastic costs

For a formal definition of the class of problems that can be tackled by *ACO/F-Race*, we follow [10]:

$$\text{Minimize} \quad F(x) = E\big[f(x,\omega)\big], \qquad \text{subject to} \quad x \in S, \tag{10.1}$$

where x is a solution, S is the set of feasible solutions, the operator E denotes the mathematical expectation, and f is the cost function which depends on x and also on a random (possibly multivariate) variable ω. The presence of the latter makes the cost $f(x,\omega)$ of a given solution x a random variable.

In the *empirical estimation* approach to stochastic combinatorial optimization, the expectation $F(x)$ of the cost $f(x,\omega)$ for a given solution x is estimated on the basis of a sample $f(x,\omega_1), f(x,\omega_2), \ldots, f(x,\omega_M)$, obtained from M independently-extracted realizations of the random variable ω:

$$\hat{F}_M(x) = \frac{1}{M}\sum_{i=1}^{M} f(x,\omega_i). \tag{10.2}$$

Clearly, $\hat{F}_M(x)$ is an *unbiased* estimator of $F(x)$.

In the case of PTSP, the elements of the general definition of a stochastic combinatorial optimization problem given above take the following meaning: A feasible solution x is an *a priori* tour visiting once and only once all cities. If cities are numbered from 1 to N, x is a permutation of $1, 2, \ldots, N$. The random variable ω is extracted from an N-variate Bernoulli distribution and prescribes which cities need being visited. In the *homogeneous* variant of PTSP, each element in ω is independently extracted from a same univariate Bernoulli distribution with probability p, where p is a parameter defining the instance. The cost $f(x, \omega)$ is the length of an *a posteriori* tour visiting the cities indicated in ω, in the order in which they appear in x.

3. The ACO/F-Race algorithm

It is straightforward to extend an *ant colony optimization* algorithm for the solution, in the *empirical estimation* sense, of a combinatorial optimization problem under uncertainty. Indeed, it is sufficient to consider one single realization of the random influence ω, say ω', and optimize the function $\hat{F}_1(x) = f(x, \omega')$. Indeed, $\hat{F}_1(x)$ is an *unbiased* estimator of $F(x)$. The risk we run by following this strategy is that we might sample a particularly *atypical* ω' which provides a misleading estimation of $F(x)$. A safer choice consists in considering a different realization of ω at each iteration of the *ant colony optimization* algorithm. The rationale behind this choice is that unfortunate modifications to the pheromone matrix that can be caused by sampling an *atypical* value of ω at a given iteration, will not have a large impact on the overall result and will be corrected in following iterations. In this paper we call *ACO-1* an *ant colony optimization* algorithm for stochastic problems in which the objective function is estimated on the basis of *one single* realization of ω which is sampled anew at each iteration of the algorithm.

A more refined approach has been proposed by Gutjahr [9, 10] and consists in using a large number of realizations for estimating the value of $F(x)$. In Gutjahr's *S-ACO* [9], the solutions produced at a given iteration are compared on the basis of a single realization. The *iteration-best* is then compared with the *best-so-far* solution on the basis of a large number of realizations. The size N_m of the sample is defined by the following equation:

$$N_m = 50 + (0.0001 \cdot n^2) \cdot k \qquad (10.3)$$

where n and k denote the size of the instance and the iteration index, respectively.

A variant of *S-ACO* called *S-ACOa* has been introduced by Gutjahr in [10] in which the size of the sample is determined dynamically on the basis of a parametric statistical test: Further realizations are considered till when either a maximum amount of computation is performed, or when the difference between the sample means for the two solutions being compared is larger than 3 times their estimated standard deviation. The selected solution is stored as the new *best-so-far* for future comparisons and is used for updating the pheromone matrix.

The *ACO/F-Race* algorithm we propose in this paper is inspired by *S-ACOa* and similarly to the latter it considers, at each iteration, a number of realizations for comparing candidate solutions and for selecting the best one which is eventually used for updating the pheromone matrix. The significant difference lies in the algorithm used at each iteration for selecting the best candidate solution. *ACO/F-Race* adopts *F-Race*, an algorithm originally developed for tuning metaheuristics [6, 5]. *F-Race* is itself inspired by a class of *racing* algorithms proposed in the machine learning literature for tackling the model selection problem [13, 14].

A detailed description of the algorithm and its empirical analysis are given in Birattari [5].

Solution methodology

The *ACO/F-Race* algorithm presents many similarities with *S-ACO* and even more with *S-ACOa* [10]. Similarly to *S-ACOa*, at each iteration it considers a number of realizations for comparing candidates solutions and for selecting the best one, which is used for updating the pheromone matrix. The sole difference between the two algorithms lies in the specific technique used to select the best candidate solution at each iteration.

In *S-ACOa*, the solutions produced at a given iteration are compared on the basis of a single realization ω to select the *iteration-best* solution. On the basis of a large sample of realizations, the size of which is computed dynamically, the *iteration-best* solution is then compared with the *best-so-far* solution. For PTSP, the solution with shorter expected *a posteriori* tour length between the two solutions is selected and stored as the new *best-so-far* solution for the subsequent iterations. This solution is used to update the pheromone matrix. In a nutshell, *S-ACOa* exploits sampling techniques and a parametric test.

ACO/F-Race employs F-Race, an algorithm based on a nonparametric test that was originally developed for tuning metaheuristics. In the context of *ACO/F-Race*, the racing procedure consists in a series of steps at each of which a new realization of ω is considered and is used for evaluating the solutions that are still in the race. At each step, a Friedman test is performed and solutions that are statistically dominated by at least another one are discarded from the race. The solution that wins the

Algorithm 1 *ACO/F-Race* Algorithm

```
input: an instance C of a PTSP problem
τ_ij ← 1, ∀(i, j)
for iteration k = 1, 2, ... do
  for ant z = 1, 2, ..., m do
    s_z ← a priori tour of ant z
  end for
  if (k = 1) then
    s_best ← F-Race(s_1, s_2, ..., s_m)
  else
    s_best ← F-Race(s_1, s_2, ..., s_m, s_best)
  end if
  τ_ij ← (1 − ρ)τ_ij, ∀(i, j)              # evaporation
  τ_ij ← τ_ij + c, ∀(i, j) ∈ s_best        # best-so-far pheromone update
end for
```

race is used for updating the pheromone and is stored as the *best-so-far*. The race terminates when either one single candidate remains, or when a maximum amount of computation time is reached.

The pseudo-code of *ACO/F-Race* is presented in Algorithm 1. The algorithm starts by initializing to 1 the pheromone on each arc (i, j) of the PTSP. At each iteration of *ACO/F-Race*, m ants, where m is a parameter, construct a solution as it is customary in ant colony optimization. In particular, we have adopted here the *random proportional rule* [8] as shown in Equation 10.4: Ant z, when in city i, moves to city j with a probability given by Equation 10.4, where N_i^z is the set of all cities yet to be visited by ant z.

$$p_{ij}^z = \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in N_i^z} \tau_{il}^\alpha \cdot \eta_{il}^\beta}, \quad if\, j \in N_i^z \qquad (10.4)$$

The m solutions generated by the ants, together with the *best-so-far* solution, are evaluated and compared via *F-Race*.

4. Experimental analysis

In the experimental analysis proposed here, we compare *ACO/F-Race* with *ACO-1*, *S-ACO* and *S-ACOa*. For convenience of the reader, we summarize here the main characteristics of the algorithms considered in this study.

ACO-1: Solutions produced at a given iteration are compared on the basis of single realization ω to select the *iteration-best* solution.

Again, on the basis of the same realization, the *iteration-best* solution is then compared with the *best-so-far* solution to select the new *best-so-far* solution.

S-ACO: Solutions produced at a given iteration are compared on the basis of a single realization ω to select the *iteration-best* solution. On the basis of a large sample of realizations, whose size is given by the equation 10.3, the *iteration-best* solution is then compared with the *best-so-far* solution.

S-ACOa: Solutions produced at a given iteration are compared on the basis of a single realization ω to select the *iteration-best* solution. On the basis of a large sample of realizations, the size of which is computed dynamically on the basis of a parametric statistical test, the *iteration-best* solution is then compared with the *best-so-far* solution.

ACO/F-Race: Solutions produced at a given iteration, together with the *best-so-far* solution, are evaluated and compared using the *F-Race* algorithm.

These four algorithms differ only for what concerns the technique used for comparing solutions and for selecting the *best-so-far* solution which is used for updating the pheromone. The implementations used in the experiments are all based on [15]. The problems considered are *homogeneous* PTSP instances obtained from TSP instances generated by the DIMACS generator [12]. We present the results of two experiments. In the first, cities are *uniformly distributed*, in the second they are *clustered*. For each of the two experiments, we consider 100 TSP instances of 300 cities. Out of each TSP instance we obtain 21 PTSP instances by letting the probability range in $[0, 1]$ with a step size of 0.05. computation time has been chosen as the stopping criterion: Each algorithm is allowed to run for 60 seconds on an AMD Opteron™ 244. These four algorithms were not fine-tuned. The parameters adopted are those suggested in [10] for *S-ACO* and are given in Table 10.1. This might possibly introduce a bias in favor of *S-ACO*. Also note that *S-ACOa* was not previously applied to PTSP. Furthermore, for PTSP, the expected cost of the objective function can be easily computed using an explicit formula given in [1]. Using this mathematical formula, the solutions selected by each algorithm on each instance were then evaluated.

In the plots given in Figures 10.1 and 10.2, the probability that cities require being visited is represented on the x-axis. The y-axis represents the expected length of the *a posteriori* tour obtained by *ACO/F-Race*,

Table 10.1. Value of the parameters adopted in the experimental analysis.

Parameter	Notation	Value
Number of ants	m	50
Pheromone exponent	α	1.0
Heuristic exponent	β	2.0
Pheromone evaporation factor	ρ	0.01
Best-so-far update constant	c	0.04

S-ACO and *S-ACOa* normalized by the expected length of the *a posteriori* tour obtained by *ACO-1*, which is taken here as a reference algorithm.

For each of the two classes of instances and for the probability values of 0.25, 0.50, 0.75, and 1.00, we study the significance of the observed differences in performance. We use the *Pairwise Wilcoxon rank sum* test [7] with p-values adjusted through Holm's method [17]. In our analysis, we consider a significance level of $\alpha = 0.01$. In Tables 10.2 and 10.3, the p-value reported at the crossing between row A and column B refers to the comparison between the algorithms A and B, where the null hypothesis is $A = B$, that is, the two algorithms produce equivalent results, and the alternative one is $A < B$, that is, A is better than B: A number smaller than $\alpha = 0.01$ in position (A, B) means that algorithm A is better than algorithm B, with confidence at least equal to $1-\alpha = 0.99$.

From the plots, we can observe that the solution quality of *ACO-1* is better than *S-ACO*, *S-ACOa* and *ACO/F-Race* for probabilities larger than approximately 0.4, that is, when the variance of the *a posteriori* tour length is small. Under such conditions, an algorithm designed to solve TSP is better than one specifically developed for PTSP. This confirms the results obtained by Rossi and Gavioli [18]. This is easily explained: Using a large number of realizations for selecting the *best-so-far* solution is simply a waste of time when the variance of the objective function is very small.

On the other hand, for probabilities smaller than approximately 0.4, the problem becomes “more stochastic”: Selecting the *best-so-far* solution on the basis of a large sample of realizations plays a significant role. The risk we run by following a single sample strategy, as in *ACO-1*, is that we might sample a particularly atypical realization which provides a misleading evaluation of solution. *S-ACO*, *S-ACOa* and *ACO/F-Race* by considering a large sample of realizations obtain better results than *ACO-1*.

Another important observation concerns the relative performance of *S-ACO*, *S-ACOa* and *ACO/F-Race*. Throughout the whole range of

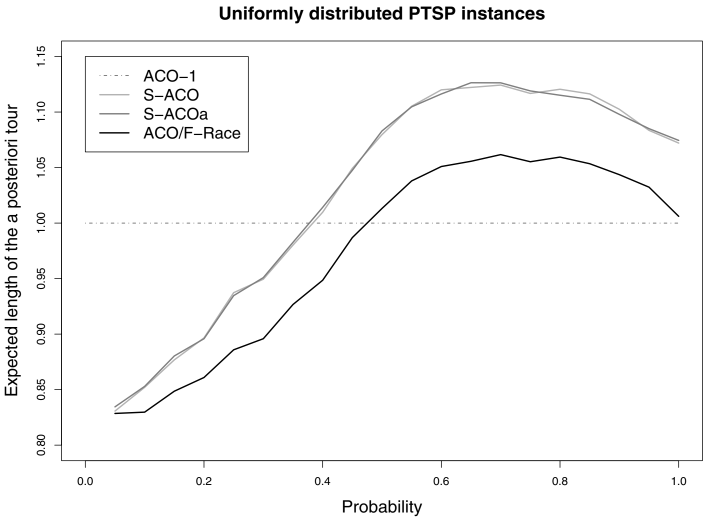

Figure 10.1. Experimental results on the uniformly distributed homogeneous PTSP. The plot represents the expected length of the *a posteriori* tour obtained by *ACO/F-Race*, *S-ACO*, and *S-ACOa* normalized by the one obtained by *ACO-1* for the computation time of 60 seconds.

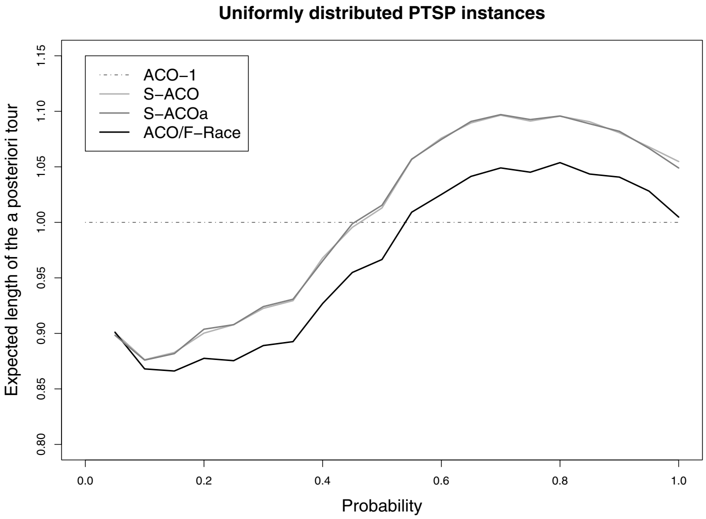

Figure 10.2. Experimental results on the clustered homogeneous PTSP. The plot represents the expected length of the *a posteriori* tour obtained by *ACO/F-Race*, *S-ACO*, and *S-ACOa* normalized by the one obtained by *ACO-1* for the computation time of 60 seconds.

probabilities, the solution quality obtained by *ACO/F-Race* is *significantly* better than the one obtained by *S-ACO* and *S-ACOa*. We can conclude that *ACO/F-Race*, with its nonparametric evaluation method, is more effective than *S-ACOa*, which uses parametric method, and than *S-ACO*, which adopts a linearly increasing sample size for selecting the *best-so-far* solution at each iteration.

In Figures 10.3 and 10.4, the average number of solutions explored by *ACO-1*, *S-ACO*, *S-ACOa* and *ACO/F-Race* is given. Since *ACO-1* uses a single realization to select the best solution, the average number of solutions explored by *ACO-1* is always larger than the those explored by *S-ACO*, *S-ACOa* and *ACO/F-Race*. Apparently a trade-off exists. The number of realizations considered should be large enough for providing a reliable estimate of the cost of solutions but at the same time it should not be too large otherwise too much time is wasted. The appropriate number of realizations depends on the stochastic character of the instance at hand. The larger the probability that a city is to be visited, the less stochastic an instance is. In this case, the algorithms that obtain the best results are those that consider a reduced number of realizations and therefore explore more solutions in the unit of time. On the other hand, when the probability that a city is to be visited is small, the instance at hand is highly stochastic. In this case, it pays off to reduce the total number of solutions explored and to consider a larger number of realizations for obtaining more accurate estimates.

In Figures 10.1 and 10.2, it should be observed that when the probability tends to 1 the curve of *ACO/F-Race* approaches 1 and therefore *ACO/F-Race* performs almost as well as *ACO-1*. This is due to the nature of the Friedman test adopted within *ACO/F-Race*. Indeed, in the deterministic case the Friedman test is particularly efficient and with a minimum number of realizations it is able to select the best solution. The computational overhead with respect to *ACO-1* is therefore relatively reduced. On the other hand, both *S-ACO* and *S-ACOa* adopt a number of realizations that is too large and therefore are able to explore only a limited number of solutions: In *S-ACO* the size of the sample does not depend on the probability and in *S-ACOa* the statistical test adopted is apparently less efficient than the Friedman test in detecting that the instance is deterministic and that a large sample is not needed. This can be observed on the far right hand side of Figures 10.1 and 10.2.

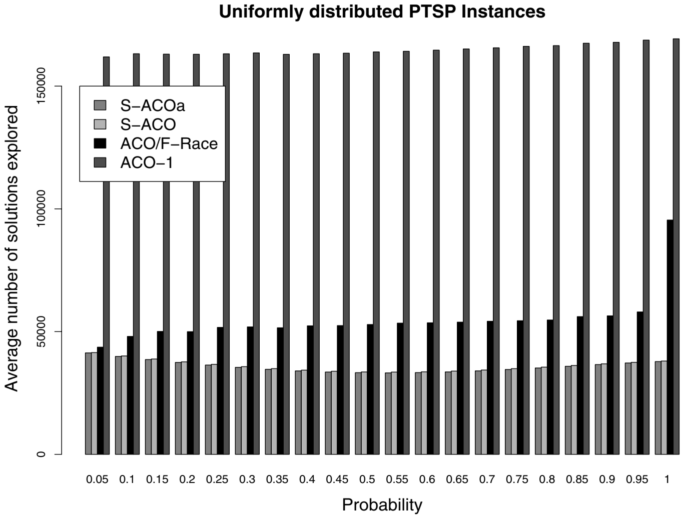

Figure 10.3. Experimental results on the uniformly distributed homogeneous PTSP. The plot represents the average number of solutions explored by *ACO-1*, *S-ACO*, *S-ACOa* and *ACO/F-Race* for the computation time of 60 seconds.

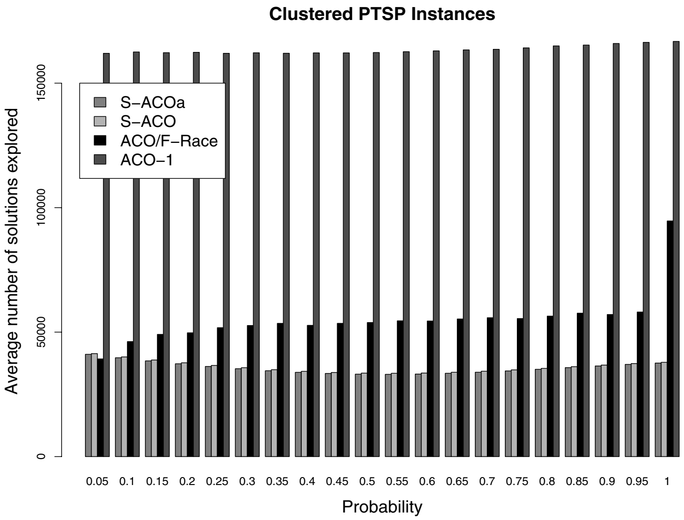

Figure 10.4. Experimental results on the clustered homogeneous PTSP. The plot represents the average number of solutions explored by *ACO-1*, *S-ACO*, *S-ACOa* and *ACO/F-Race* for the computation time of 60 seconds.

Table 10.2. The p-values of the paired Wilcoxon tests on uniformly distributed homogeneous PTSP instances. The quantities under analysis are the expected length of the *a posteriori* tour obtained by *ACO/F-Race*, *S-ACO*, *S-ACOa* and *ACO-1*.

Probability=0.25	*ACO/F-Race*	*S-ACO*	*S-ACOa*	*ACO-1*
ACO/F-Race	–	$< 2.2e-16$	$< 2.2e-16$	$< 2.2e-16$
S-ACO	1	–	1	$< 2.2e-16$
S-ACOa	1	$< 2.2e-16$	-	$< 2.2e-16$
ACO-1	1	1	1	–
Probability=0.5	*ACO/F-Race*	*S-ACO*	*S-ACOa*	*ACO-1*
ACO/F-Race	–	$< 2.2e-16$	$< 2.2e-16$	1
S-ACO	1	–	$< 2.2e-16$	1
S-ACOa	1	1	–	1
ACO-1	$< 2.2e-16$	$< 2.2e-16$	$< 2.2e-16$	–
Probability=0.75	*ACO/F-Race*	*S-ACO*	*S-ACOa*	*ACO-1*
ACO/F-Race	–	$< 2.2e-16$	$< 2.2e-16$	1
S-ACO	1	–	$< 2.2e-16$	1
S-ACOa	1	1	–	1
ACO-1	$< 2.2e-16$	$< 2.2e-16$	$< 2.2e-16$	–
Probability=1.0	*ACO/F-Race*	*S-ACO*	*S-ACOa*	*ACO-1*
ACO/F-Race	–	$< 2.2e-16$	$< 2.2e-16$	1
S-ACO	1	–	$< 2.2e-16$	1
S-ACOa	1	1	–	1
ACO-1	$< 2.2e-16$	$< 2.2e-16$	$< 2.2e-16$	–

Table 10.3. The p-values of the paired Wilcoxon tests on clustered homogeneous PTSP instances. The quantities under analysis are the expected length of the *a posteriori* tour obtained by *ACO/F-Race*, *S-ACO*, *S-ACOa* and *ACO-1*.

Probability=0.25	*ACO/F-Race*	*S-ACO*	*S-ACOa*	*ACO-1*
ACO/F-Race	–	$< 2.2e-16$	$< 2.2e-16$	$< 2.2e-16$
S-ACO	1	–	0.6845	$< 2.2e-16$
S-ACOa	1	< 0.3155	–	$< 2.2e-16$
ACO-1	1	1	1	–
Probability=0.5	*ACO/F-Race*	*S-ACO*	*S-ACOa*	*ACO-1*
ACO/F-Race	–	$< 2.2e-16$	$< 2.2e-16$	$< 2.2e-16$
S-ACO	1	–	$< 2.2e-16$	1
S-ACOa	1	1	–	1
ACO-1	1	$< 2.2e-16$	$< 2.2e-16$	–
Probability=0.75	*ACO/F-Race*	*S-ACO*	*S-ACOa*	*ACO-1*
ACO/F-Race	–	$< 2.2e-16$	$< 2.2e-16$	1
S-ACO	1	–	$< 2.2e-16$	1
S-ACOa	1	1	–	1
ACO-1	$< 2.2e-16$	$< 2.2e-16$	$< 2.2e-16$	–
Probability=1.0	*ACO/F-Race*	*S-ACO*	*S-ACOa*	*ACO-1*
ACO/F-Race	–	$< 2.2e-16$	$< 2.2e-16$	1
S-ACO	1	–	1	1
S-ACOa	1	$< 2.2e-16$	–	1
ACO-1	$< 2.2e-16$	$< 2.2e-16$	$< 2.2e-16$	–

5. Conclusions and future work

The preliminary experimental results proposed in Section 4 confirm the generality of the approach proposed by Gutjahr [9, 10], and show that the *F-Race* algorithm can be profitably adopted for comparing solutions in the framework of the application of *ant colony optimization* to combinatorial optimization problems under uncertainty.

Further research is needed for properly assessing the quality of the proposed *ACO/F-Race*. We are currently developing an *estimation*-based local search for PTSP. We plan to study the behavior of *ACO/F-Race* enriched by this local search on *homogeneous* and *non-homogeneous* problems.

In the experimental analysis proposed in Section 4, the goal was to compare the evaluation procedure based on *F-Race* with the one proposed in [10] and with the trivial one based on a single realization. For this reason, solution construction and pheromone update were implemented as described in [9, 10]. We plan to explore other possibilities, such as construction and update as defined in $\mathcal{MAX}$–$\mathcal{MIN}$ *ant system* [16]. Applications to other problems, in particular of the VEHICLE ROUTING class, will be considered too.

Acknowledgments. This research has been supported by COMP^{2}SYS, a Marie Curie Early Stage Research Training Site funded by the European Community's Sixth Framework Programme under contract number MEST-CT-2004-505079, and by the ANTS project, an *Action de Recherche Concertée* funded by the Scientific Research Directorate of the French Community of Belgium. Marco Dorigo acknowledges support from the Belgian FNRS, of which he is a Research Director.

References

[1] D. J. Bertsimas, P. Jaillet, and A. Odoni. A priori optimization. *Operations Research*, 38:1019–1033, 1990.

[2] L. Bianchi. *Ant Colony Optimization and Local Search for the Probabilistic Traveling Salesman Problem: A Case Study in Stochastic Combinatorial Optimization.* PhD thesis, Université Libre de Bruxelles, Brussels, Belgium, 2006.

[3] L. Bianchi, L. M. Gambardella, and M. Dorigo. Solving the homogeneous probabilistic travelling salesman problem by the ACO metaheuristic. In M. Dorigo, G. Di Caro, and M. Sampels, editors, *Ant Algorithms, 3rd International Workshop, ANTS 2002*, volume 2463 of *LNCS*, pages 176–187, Berlin, Germany, 2002. Springer-Verlag.

[4] M. Birattari. Race. R package, 2003. `http://cran.r-project.org`.

[5] M. Birattari. *The Problem of Tuning Metaheuristics as Seen from a Machine Learning Perspective.* PhD thesis, Université Libre de Bruxelles, Brussels, Belgium, 2004.

[6] M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp. A racing algorithm for configuring metaheuristics. In W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 11–18, San Francisco, CA, USA, 2002. Morgan Kaufmann.

[7] W. J. Conover. *Practical Nonparametric Statistics.* John Wiley & Sons, New York, NY, USA, third edition, 1999.

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

[9] W. J. Gutjahr. A converging ACO algorithm for stochastic combinatorial optimization In A. Albrecht, and T. Steinhöfl, editors, *Proc. SAGA 2003*, volume 2827 of *LNCS*, pages 10–25, Berlin, Germany, 2003. Springer-Verlag.

[10] W. J. Gutjahr. S-ACO: An ant based approach to combinatorial optimization under uncertainity. In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Stützle, editors, *Ant Colony Optimization and Swarm Intelligence, 4th International Workshop, ANTS 2004*, volume 3172 of *LNCS*, pages 238–249, Berlin, Germany, 2004. Springer-Verlag.

[11] P. Jaillet. *Probabilistic Travelling Salesman Problems.* PhD thesis, The Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.

[12] D. S. Johnson, L. A. McGeoch, C. Rego, and F. Glover. 8th DIMACS implementation challenge.
`http://www.research.att.com/ dsj/chtsp/`, 2001.

[13] O. Maron and A. W. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 59–66, San Francisco, CA, USA, 1994. Morgan Kaufmann.

[14] A. W. Moore and M. S. Lee. Efficient algorithms for minimizing cross validation error. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 190–198, San Francisco, CA, USA, 1994. Morgan Kaufmann.

[15] T. Stützle. ACOTSP, version 1.0. `http://www.aco-metaheuristic.org/aco-code/`, 2002.

[16] T. Stützle and H. H. Hoos. $\mathcal{MAX}$–$\mathcal{MIN}$ ant system. *Future Generation Computer Systems*, 16(8):889–914, 2000.

[17] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*,volume 6, 65–70,1979.

[18] F. Rossi and I. Gavioli. Aspects of heuristic methods in the probabilistic traveling salesman problem. *Advanced School on Stochastics in Combinatorial Optimization.* pages 214–227. World Scientific, Hackensack, NJ, USA, 1987.

[19] S. M. Weiss and C. Kulikowski *Computer systems that learn. Classification and prediction methods from statistics neural nets machine learning and expert systems.* Morgan Kaufmann, San Mateo, CA, USA, 1991.