2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Parameter Optimization by Means of Statistical Quality Guides in F-Race

R. Klazar and A.P. Engelbrecht

Abstract— **F-Race and its variant, Iterated F-Race, is an automated procedure for sampling and evaluating potential values of parameters for algorithms. The procedure is controlled by means of a computational budget that limits the number of evaluations that may be conducted, thus forcing the determination of the best possible configuration to be made within a limited time. When time is not severely constrained, the *a priori* choice of a computational budget becomes unjustifiable because the relationship between the computational budget and the quality of the optimization of a black box subject is not obvious. This paper proposes an extension to F-Race in the form of a heuristic method for reasonably terminating the optimization procedure.**

I. Introduction

THE application of a metaheuristic to a problem includes the configuration of the metaheuristic by the choosing of its variable components and the values of its free parameters. When no suitable configurations are known it becomes necessary to search for a reasonable, if not optimal, configuration by means of a structured procedure. The specific procedure may vary, depending on the nature of the problem and the presence or absence of expert, domain knowledge, but generally involves the selection and evaluation of candidate configurations. The goal of an optimization procedure is to locate a space of reasonable configurations or the best configuration amongst a fixed-size set of candidate configurations.

When the number of free parameters is low, the possible parameter values are discrete, and the search space is small or finite, a brute force approach, whereby every combination of values is tested, is a feasible choice of optimization procedure. However, when the number of free parameters increases, the possible parameter values are continuous, and the space of values is large or infinite, a linear search can become too time-consuming to be feasible. Equally so, the metaheuristic to be optimized can require a non-trivial amount of time to complete a run, thereby contributing to an overall impractical total time to run the search for suitable parameter values.

A commonly used alternative method is that of *full factorial design* (FFD) [9]. An optimization procedure that employs FFD will sample values at varying levels, l, for the free parameters (factors), f, to yield l^f configurations. Initially, the levels are chosen with some expert knowledge of the domain. Once the initial set of samples is evaluated, the search can be refined by focusing on a promising area of the search space, as determined by the testing of the initial samples. New values are then once again sampled according to levels that are determined by the domain expert. Thus is the optimization guided and eventually terminated by the researcher.

F-Race [3][4] is a structured procedure that facilitates the automation of the selection and evaluation of candidate configurations and provides a specific criterion by which superior candidates are chosen, namely, statistical significance tests. Nevertheless, F-Race leaves open the choices of the initial set of candidates, the number of tests to perform (alternatively, the duration of testing), and the number of elite, surviving candidates from which to select a suitable configuration or by which to guide further optimization.

Iterated F-Race [2] is an extension to F-Race that refines the search for configurations by narrowing the configuration search space over multiple iterations. Each iteration is a run of the F-Race algorithm. The candidate configurations for the initial iteration are chosen by the researcher. The candidate configurations for subsequent iterations are automatically sampled around one of an elite group of surviving candidates from a previous iteration.

This paper presents two modifications to F-Race with the aim of producing a variant algorithm that is not dependent on parameters as much as on a well-defined procedure. Furthermore, the new variant is intended to be used in situations where time is not extremely constrained and exploration of a configuration search space is both feasible and desired. Simultaneously, unneccessary computations should still be avoided by stopping the optimization procedure when configurations of a reasonable quality have been found and further exploration would become costly.

In related research, Yuan *et al.* [15] have studied the use of mesh adaptive direct search (MADS) to control the number of evaluations of a candidate configuration adaptively and have produced a hybrid of MADS and F-Race. The MADS/F-Race algorithm allocates more computing time to promising candidates and less to inferior ones. Branke and Elomari [6] have proposed setting the significance level of the statistical significance tests employed by F-Race adaptively in order to eliminate the need to choose the level manually, thus eliminating a free parameter of the F-Race procedure.

The remainder of this paper presents each modification in detail and is organized as follows: Section II provides an overview of F-Race and Iterated F-Race, Section III describes the proposed modifications, Section IV describes the experimental procedure used to test the modifications and discusses the results, and Section V concludes the paper

Ronald Klazar and Andries Engelbrecht are with the Department of Computer Science, University of Pretoria, Pretoria, South Africa (email: {rklazar, engel}@cs.up.ac.za).

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

TABLE I

VISUALIZATION OF CANDIDATE CONFIGURATIONS, PROBLEMS, AND COSTS IN F-RACE. P IS THE PROBLEM CLASS, $P_1, \ldots, P_k$ ARE THE PROBLEM INSTANCES, $\theta_1, \ldots, \theta_m$ ARE THE CANDIDATE CONFIGURATIONS, AND $c_i^{\theta_j}$ IS THE COST OF APPLYING THE SUBJECT WITH CONFIGURATION θj TO PROBLEM INSTANCE P_i FOR $i = 1, \ldots, k$ AND $j = 1, \ldots, m$.

P	θ_1	θ_2	$\ldots$	θ_m
P_1	$c_1^{\theta_1}$	$c_1^{\theta_2}$	$\cdots$	$c_1^{\theta_m}$
P_2	$c_2^{\theta_1}$	$c_2^{\theta_2}$	$\cdots$	$c_2^{\theta_m}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
P_k	$c_k^{\theta_1}$	$c_k^{\theta_2}$	$\cdots$	$c_k^{\theta_m}$

with a summary of the findings and suggested directions for further study.

II. OVERVIEW OF F-RACE AND ITERATED F-RACE

F-Race repeatedly evaluates a set of candidate configurations in specifying an application of a metaheuristic, the subject, to a class of problems. Each evaluation of a candidate yields evidence of its fitness for configuring the subject. When sufficient evidence is gathered to differentiate the candidates statistically, the lower performing candidates are discarded. Evaluation continues until either a specified number of candidates remains or a predetermined computational budget is exhausted.

A problem class describes multiple problem instances, where instances can differ by such variations as stochastic perturbations and inputs from an unpredictable operating environment. In such a case, it is not feasible to evaluate the entire population of the problem class in order to find the configuration that is optimal over all instances. Instead, a subset of problem instances is chosen to represent the problem class such that a generalization about the fitness of a configuration can be made. For instance, if the testing problem subset is chosen according to the same probability model as that by which problem instances occur in practice, then it can be justifiably argued that the testing problem subset represents the problem class.

F-Race proceeds in steps. At each step, each candidate configuration in the current set is applied to the same problem instance. The candidates are then ranked in terms of their fitness. The measure of fitness depends on the optimization criterion for the specific problem class. Table I depicts the arrangement of candidates, problem instances, and fitness values.

Comparing candidates in pairs for a statistically significant difference in fitness is computationally wasteful if conducted at each step, regardless of the existence of sufficient evidence to discard any of the candidates. Therefore, a single test of the entire set of candidates is conducted first and if this initial test suggests that a statistically significant difference does exist within the current set of candidates, then pairs of candidates are tested in order to determine which candidates should be discarded. F-Race employs the Friedman test for variance by ranks [7] to test a candidate set, and the Wilcoxon signed ranks test [7] to test candidate pairs.

Candidates that remain after F-Race is stopped are the best configurations in the initial candidate set but are not necessarily the best existing configurations. Additional testing may be required to locate a better candidate or to reject, with some certainty, the existence of a better candidate. Iterated F-Race defines the procedure for choosing a new set of candidates that are based around an elite, surviving candidate from the previous run of the F-Race procedure. Repeated iterations of F-Race narrow the search space of candidate configurations by using the best performing candidates from previous iterations as a guide.

Birattari *et al.* [5] suggest choosing an elite set of survivors by $N_e = min(N_{survive}, N_{min})$ where $N_{survive}$ is the number of candidates remaining after the previous iteration of F-Race completes and N_{min} is a predetermined, desired number of survivors. The elite candidates are weighted by

$$w_z = \frac{N_e - r_z + 1}{N_e \cdot (N_e + 1)/2} \tag{1}$$

for $z = 1, \ldots, N_e$ and where r_z is the rank of an elite configuration. One of the elite survivors is then chosen with a probability that is proportional to w_z and each new candidate configuration, $x = (x^1, x^2, \ldots, x^m)$, is sampled around the chosen elite candidate, $x_z = (x_z^1, x_z^2, \ldots, x_z^m)$, where m is the number of parameters. Each component, x^i, is sampled according to a normal distribution with x_z^i as the mean and σ_l^i as the standard deviation, defined by

$$\sigma_{l+1}^i = v^i \cdot \left(\frac{1}{N_{l+1}}\right)^{\frac{1}{d}} \tag{2}$$

for $l = 1, \ldots, L-1$ and where L is the number of iterations, d is the number of components of a configuration, and v^i is the range of the component x^i. The elite candidate is included with the new newly sampled candidates and all of these candidates are tested again. The implication of this design is that the bias of the sampling distribution towards the elite candidate is increased as the number of components (or parameters) is increased and as the number of candidate configurations to be sampled is increased.

The computational budget, B, is distributed over all iterations according to

$$B_l = \frac{B - B_{used}}{L - l + 1} \tag{3}$$

where $l = 1, \cdots, L$ and B_{used} is the computational budget used up to and including iteration $l - 1$.

Each iteration of F-Race is stopped when at most N_{min} candidate configurations remain, where

$$N_{min} = 2 + round(log_2 d) \quad (4)$$

The choice of the candidate set for the first iteration, the conditions under which each iteration is terminated, and the method by which subsequent candidate configurations are selected, as described above, are modified in this paper. The following section discusses the proposed modifications in detail.

III. Modifications to the F-Race and Iterated F-Race Procedures

F-Race is designed to balance computational time with quality of optimization. However, the duration of the optimization procedure is chosen by the researcher. When the subject to be optimized is a black box or exploration of the configuration search space is desired, a heuristic that is relatable to the quality of the optimization presents a justifiable means of controlling the expenditure of computational time. The modifications presented next aim to replace free parameters with non-configurable procedures without affecting the general quality of optimization.

A. Initial Candidate Selection

The initial set of candidate configurations determines the extent of the search space within which a good or optimal configuration is expected to be found. A single iteration of F-Race will select the most promising one (or more) candidates from a set. A subsequent iteration of F-Race will generate new candidates around the most promising candidate from the previous iteration and select the most promising candidate from the new set, thus narrowing the extent of the search space and increasing the resolution of the search.

The choice of initial candidates depends on a researcher's insight into the subject to be optimized. Domain knowledge of the subject can be used to determine the range of values for each configuration parameter; expertise can inform the choice of parameter levels in a full factorial design. Whether or not the initial search space can be constrained, the location of possibly good configurations within the search space is not exactly known.

One approach to initiating the search is to sample the initial candidate configurations randomly from the defined search space. The choice of random number generator is a determining factor in the distribution of this sample. Figure 1 depicts a distribution of configurations, of two parameters each, sampled using the widely known Mersenne Twister [12] pseudo-random number generator. Ideally, the search space should be covered as evenly as possible to avoid leaving unexplored areas and to avoid clustering the initial configurations, thus wasting time on comparing relatively similar configurations when exploration is desired. In the case of the Mersenne Twister, relatively large spaces can be left unsampled while some configurations are sampled close together. Figure 2 depicts a more evenly distributed set of configurations that were sampled using a Sobol sequence

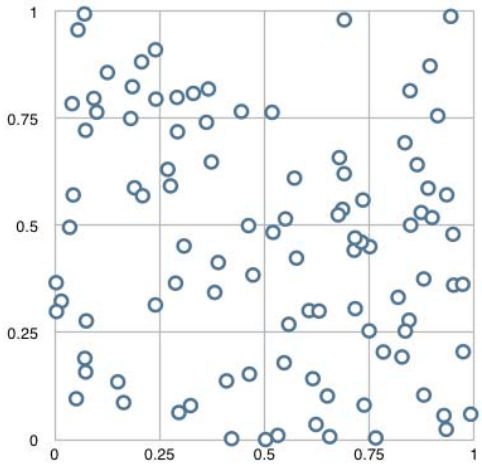

Fig. 1. 100 points sampled using the Mersenne Twister pseudo-random number generator. A primary generator was used to produce seeds for the secondary generators, each of which produced a sequence for one of the parameters. (Primary seed was 141141652.)

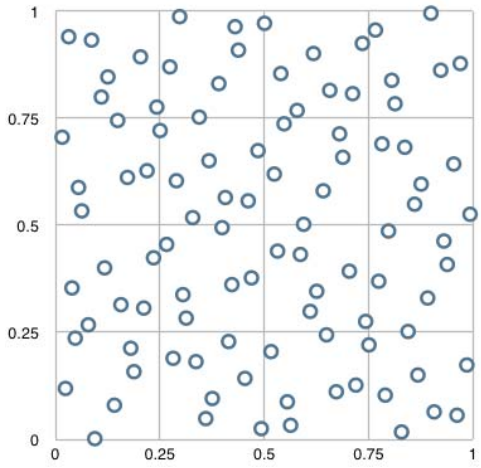

Fig. 2. 100 points sampled using the Sobol quasi-random number generator. (Skipped initial 4096 points.)

[13]. As noted by Joe and Kuo [11], it has been suggested that an initial part of the sequence is skipped in order to compensate for a potentially poor choice of the initial direction numbers used to initialize the Sobol generator. Acworth, Broadie, and Glasserman [1] suggest skipping the largest power of two that is smaller than the number of points desired. Since this study experimented with a variety of sample sizes, none of which exceed 10^4, the skip count was fixed at 4096 points for the sake of consistency.

When the distribution of candidates is uneven, an increased number of candidates is required to cover the search space. On the other hand, an even distribution of candidates allows for the first F-Race iteration to test lower numbers of configurations in order to determine which areas within the search space are promising. As such, a Sobol quasirandom number generator is proposed as the sampling method for

cases where a random sample of a search space is required.

B. Termination Condition

The proposal of a novel metaheuristic or a new variation or application of an existing metaheuristic necessitates the substatiation of the choice of the configuration employed. A substantiation can be a description of the procedure followed in determining the configuration or a reference to a consensus on an established, reasonable configuration in the existing literature. F-Race provides a well-defined procedure.

When the time available for optimization is limited, a computational budget serves as a means to find the best possible configuration within said time constraints. Furthermore, when the subject to be optimized is well understood and the initial set of configuration candidates is chosen accordingly, a computational budget limits unnecesary testing that would otherwise be conducted beyond an expected outcome of the optimization procedure. Nevertheless, the choice of a computational budget and that of a termination condition can be difficult to justify.

When the time available for optimization is not severely limited, it is preferable to allow the optimization procedure to explore the given search space. When the subject to be optimized is a black box, one cannot know the number of evaluations of candidate configurations that will be required to differentiate the configurations to a statistically significant degree. The uncertainty in the choice of a computational budget is due to it not being possible to state reasonably that exceeding the budget is likely to be fruitless and, likewise, that the budget is not unnecessarily excessive and therefore wasteful.

F-Race is based on the use of the Friedman test for variance by ranks to determine if a statistically significant difference in fitness exists within a set of candidate configurations. This test is employed to avoid testing each pair of candidates when no statistically significant difference is expected to be found and is conducted at each step of the optimization procedure - in other words, each time that the set of candidates is evaluated against a new problem instance. This paper proposes using the p-values of the Friedman test as an indicator of the likelihood that statistically significant differences will be found in further testing.

A set of candidate configurations would typically be evaluated against multiple problems before a statistically significant difference between them could be established. Birattari *et al.* [5] refer to this as "gathering evidence". In cases where a difference between candidates is to be found, the p-values produced by the Friedman test depict a downward trend, which eventually terminates in a sufficiently low p-value to effect a discard of one or more of the candidates. Naturally, in cases where the candidates can no longer be differentiated after any number of evaluations the trend of Friedman test p-values would be observed eventually to become constant.

The proposed method for terminating the optimization procedure is to gather a minimum number of Friedman test p-values as evidence of the trend of the evaluation results. The trend is determined by means of least squares linear regression, whereby a line is fitted to the sample of p-values. When one or more candidates are discarded, the sample of p-values is discarded and a new sample is taken. As long as no candidates are discarded and the slope of the trend line is decreasing, optimization is permitted to continue. Once the slope of the line becomes constant or begins to increase, optimization is terminated by the reason that testing no longer appears to lead towards further discarding of candidates.

The minimum size of the p-value sample is a constant and should not be used as a parameter. The number of p-values sampled before a trend line is calculated is related to a computational budget in that changes to the minimum sample size would be subject to the same *a priori* estimates. For this proposal, a minimum of ten p-values are sampled before the termination condition is evaluated.

The size of the p-value sample is increased with each new test as long as no candidates are discarded. An alternative was considered whereby the sample size n would remain fixed and the sample would be a sliding window that would include the latest n p-values. However, this method would introduce a parameter that would subject the outcome of an optimization procedure to a potentially arbitrary choice of value for n. Instead, the outcome remains dependant on the quality of the underlying procedure.

While a least squares fit (LSF) was used in this proposal, because the method is sensitive to outliers a more robust approach like least absolute deviations (LAD) may be more economical. At issue is the typical progression of p-values, which start out relatively large and tend to decrease somewhat either before reaching a critical value or stabilising at some higher value. The slope of an LSF trend line is affected by the initial, large p-values and many smaller, subsequent p-values are required to alter that slope. Such situations would incur unnecessary testing before termination. An LAD trend line would not be affected by outliers as much as the LSF trend line and might be a more suitable method.

IV. Experiments and Results

The modified version of the F-Race algorithm, herein referred to as Heuristic F-Race (H/F-Race), was compared with the unmodified version (F-Race) in terms of the quality of the configurations found and the number of function evaluations performed to find a configuration. The subject provided for parameter optimization was an application of the ant system (AS) [8] version of the ant colony optimization (ACO) metaheuristic to the travelling salesman problem. Both algorithm and problem were chosen because they are simple and easy to implement. Problem instances were obtained from the *pla85900* dataset of TSPLIB95 [14], which defines a fully connected graph of 85900 nodes. Each problem instance consisted of a randomly chosen subset of 100 of the total nodes. The problem class was thus all possible combinations of 100 nodes. Problem instances were generated randomly during optimization and each instance was unique. Configurations were tested on problem instances

generated in the same way and excluded all problem instances used during optimization.

Five parameters of the AS algorithm were chosen for optimization: pheromone influence, heuristic information influence, pheromone evaporation rate, initial pheromone, and number of ants, which in [8] are α, β, ρ, τ_0, and m, respectively. The ranges of values from which initial candidates were generated were informed by the suggested settings for ACO algorithms without local search in [8] and are listed in Table II.

TABLE II
RANGES OF PARAMETER VALUES FOR INITIAL CONFIGURATION CANDIDATES

Parameter	Range
α	[0.0,1.0]
β	[0.0,10.0]
ρ	[0.0,1.0]
τ_0	[0.0,1.0]
m	[1,100]

Furthermore, each run of the AS algorithm with a candidate configuration and a problem instance was allowed to perform 25 iterations and to execute to completion. The distance of the shortest path reported after the 25 iterations were completed was taken as the fitness value by which the candidate configuration was evaluated.

To establish the performance of H/F-Race in relation to that of the unmodified algorithm, the modified algorithm was executed first to produce 30 runs. The mean number of function evaluations per run was calculated and this value was used as the basis for a choice of budget for the unmodified algorithm. The unmodified algorithm was executed for 30 runs with only the minimum budget required to complete an evaluation, then again with half the mean number of function evaluations as budget, and, finally, with double the mean number of function evaluations as budget. In total, thirty candidate configurations were sampled for each execution of each algorithm. The configurations found by H/F-Race were then compared with those found by the three executions of the unmodified algorithm. If multiple candidate configurations survived after an optimization run, then the candidate with the lowest rank was chosen as the final configuration.

To evaluate the found configurations, each set of configurations found by each algorithm (and budget variation) was evaluated in an application of AS to 100 new problem instances that did not include any of the instances used during optimization. The mean of the shortest path length over each candidate's 30 runs was calculated to produce 30 means for each algorithm and budget variation. These sets of 30 means were then compared using the Student's t-test to determine if there was a statistically significant difference between the results obtained by each algorithm.

The Friedman test statistic is approximated by the χ^2 distribution when the number of ranks (candidate configurations) is three and the number of observations (problem instances) is greater than nine and when the number of ranks is four and the number of observations exceeds four [10]. The implementation of F-Race used for this study evaluated 10 problem instances before beginning testing in order to ensure a good approximation by the Friedman statistic of χ^2. Therefore, the minimum number of evaluations performed before statistical testing was begun was ten times the number of configuration candidates.

The significance level used for the Friedman test by each algorithm was set to 95.0% and that of the Wilcoxon signed ranks test was set to 97.5%.

No limit on the number of surviving candidates was set and each run was terminated when the computation budget was exhausted, when the termination heuristic determined that no further testing should be conducted, or when only one configuration remained.

The size of a candidate configuration set was chosen as follows. Each dimension of the configuration search space was divided into some number of segments. Each segment was required to contain at least one configuration. Each configuration that appeared in a segment of one dimension, was required to appear in a segment of each of the other dimensions. Candidate configurations were then generated by the Sobol sequence generator until no segments were left empty.

The experiment divided the search space, as defined by Table II, according to the segment sizes $0.33, 3.30, 0.33, 0.33, 33.00$ for the parameters, α, β, ρ, τ_0, and m, respectively, to produce 1750 candidate configurations. As described previously, the minimum number of function evaluations performed by both algorithms was 17500, after which statistical testing commenced. The mean number of function evaluations performed by H/F-Race after the initial 17500 evaluations was approximately 4939. Therefore, the zero-budget execution was allocated a budget of 17500, the half-budget execution was allocated a budget of 19970, and the double-budget execution was allocated a budget of 27378. The results of the experiment are presented in Table III and Table IV, with *H/F* denoting H/F-Race and *F* denoting F-Race.

TABLE III
P-VALUES FOR COMPARISONS OF RESULTING CONFIGURATIONS

Algorithm (budget)	**F (0.0)**	**F (0.5)**	**H/F (1.0)**	**F (2.0)**
H/F-Race (1.0)	**0.0294**	0.1808	-	-
F-Race (0.5)	0.3129	-	-	-
F-Race (2.0)	**0.0183**	0.1202	0.8164	-

The resulting mean shortest path values are in line with expectations that larger budgets afford the general F-Race algorithm more opportunity to determine the best configuration from the given initial set of candidates. The results of the statistical significance tests, interpreted at a significance level of 95%, suggest that the zero-budget execution leaves room for further gains to be made with a larger budget.

The hybrid algorithm obtained significantly better config-

TABLE IV

SHORTEST PATHS FOR COMPARISONS OF RESULTING CONFIGURATIONS, ORDERED BY MEAN VALUE.

Algorithm (budget)	Mean	Std Deviation
F-Race (0.0)	6046091.5733	41232.4964
F-Race (0.5)	6035970.1900	34151.1044
H/F-Race (1.0)	6023675.4967	34969.4114
F-Race (2.0)	6021512.2867	35654.7412

urations than the zero-budget execution but not more so than the half-budget execution. However, the half-budget execution did not obtain significantly better configurations than the zero-budget execution. This suggests that the computing time spent by the hybrid algorithm was not wasted and illustrates that should a researcher have provided an estimated budget near to that of the half-budget execution, the optimization procedure would have been terminated too soon and would have been a waste of computing time.

The double-budget execution did not signficiantly improve upon the results of the hybrid algorithm, suggesting that there was little to be gained beyond the termination point determined by the heuristic. The latter statement is supported furthermore by the relatively large p-value obtained by the comparison of the hybrid algorithm with the double-budget execution. Once again, the implication is that an estimated budget, introduced as a parameter value to the configuration procedure, and falling beyond the termination point determined by the heuristic would have yielded little gain in better configurations.

V. CONCLUSION

This paper presented an extension to the F-Race optimization procedure that aims to alter the algorithm's locus of control in favour of exploration without specific cognisance to a computational budget. The duration of the exploration is left to a heuristic that does not need to be configured by a researcher and can be referenced in the substantiation of a statement of parameters in an empirical study.

The modified F-Race algorithm was evaluated with a focus on the heuristic employed to decide when optimization should be stopped. This was done to determine the viability of the principle idea. While the results suggest that the heuristic can balance function evaluation with optimization quality, the results do not rule out the possibility that the heuristic will terminate an optimization procedure before the best possible configurations are found.

Therefore, future work will aim to ascertain the volatility of the LSF method employed by this study and to test the LAD regression method as an alternative heuristic.

REFERENCES

[1] P. A. Acworth, M. Broadie, P. Glasserman, "A comparison of some Monte Carlo and quasi-Monte Carlo techniques for option pricing," *Monte Carlo and Quasi-Monte Carlo Methods 1996, H. Niederreiter, P. Hellekalek, G. Larcher, and P. Zinterhof, Eds. Lecture Notes in Statistics*, vol. 127, pp. 1-18, Springer, New York, 1996.

[2] P. Balaprakash, M. Birattari, and T. Stützle, "Improvement strategies for the F-Race algorithm: Sampling design and iterative refinement," *Hybrid Metaheuristics, 4th International Workshop, HM 2007, Lecture Notes in Computer Science*, vol. 4771, pp. 108122, Springer Verlag, Berlin, Germany, 2007.

[3] M. Birattari, T. Stützle, L. Paquete, K. Varrentrapp, "A Racing Algorithm for Configuring Metaheuristics," *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002*, pp. 11-18, San Francisco, Morgan Kaufmann, 2002.

[4] M. Birattari, "The Problem of Tuning Metaheuristics as Seen from a Machine Learning Perspective," PhD thesis, Universit'e Libre de Bruxelles, Brussels, Belgium, 2004.

[5] M. Birattari, Z. Yuan, P. Balaprakash, T. Stützle, "F-Race and iterated F-Race: An overview," *Experimental methods for the analysis of optimization algorithms*, pp. 311-336. Springer, Berlin, 2010.

[6] J. Branke and J. Elomari, "Racing with a Fixed Budget and a Self-Adaptive Significance Level," *Learning and Intelligent Optimization, 7th International Conference, LION 7, Lecture Notes in Computer Science*, pp. 272-280, 2013.

[7] W.J. Conover, *Practical Nonparametric Statistics.* John Wiley & Sons, 3rd edition, 1999.

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

[9] R. A. Fisher *The Design of Experiments.* Oliver & Boyd, Oxford, England, 1935.

[10] M. Friedman, "The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675-701, 1937.

[11] S. Joe and F. Y. Kuo, "Remark on Algorithm 659: Implementing Sobols Quasirandom Sequence Generator," *ACM Transactions on Mathematical Software*, vol. 29, no. 1, pp. 4957, 2003.

[12] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS) - Special issue on uniform random number generation*, vol. 8, no.1, pp. 3-30, 1998.

[13] I. M. Sobol, "On the distribution of points in a cube and the approximate evaluation of integrals," *U.S.S.R. Computational Mathematics and Mathematical Physics*, vol. 7, no. 4, pp. 86112, 1967.

[14] *TSPLIB* http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

[15] Z. Yuan, T. Stützle, M. Birattari, "MADS/F-race: mesh adaptive direct search meets F-race," *Trends in Applied Intelligent Systems*, Springer Berlin Heidelberg, pp. 41-50, 2010.