Parameter Tuning: F-Race

F-Race [Birattari02] – procedure that empirically evaluates a set of candidate configurations by discarding bad ones as soon as statistically sufficient evidence is gathered against them.

- The process starts from a given finite pool of candidate configurations.
- If sufficient evidence is gathered that some candidate is inferior to at least another one, such a candidate is dropped from the pool and the procedure is iterated over the remaining ones.

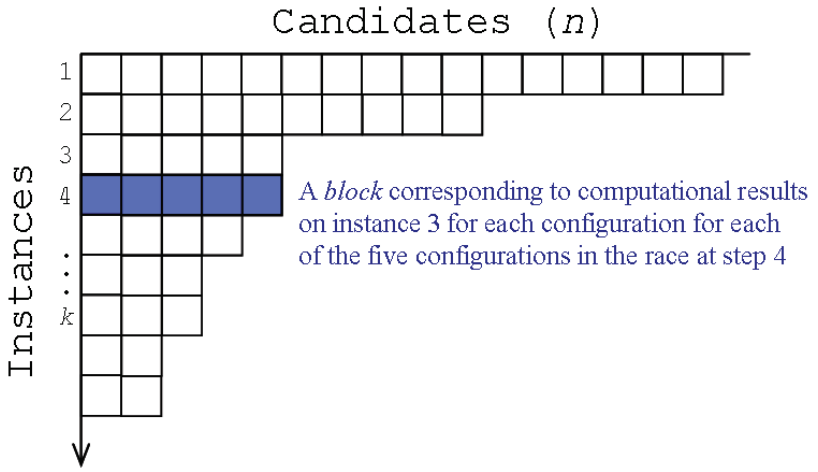
The methodology can be applied to repetitive problems – problems where many similar instances appear over time.

F-Race: Algorithm

- k is the current step of the race process and $n = |\Theta_{k-1}|$ configurations are still in the race.
- $\underline{i}$ is a random sequence of training instances; $\underline{i}_k$ is drawn from I according to P_I, independently for each k.
- $\underline{c}^k(\theta, \underline{i})$ is an array of k terms; $c(\theta, \underline{i}_l)$ is the cost of the best solution found by configuration θ on instance $\underline{i}_l$.

  For a given θ, the array $\underline{c}^k$ of length k can be obtained from $\underline{c}^{k-1}$ by appending the cost concerning the k-th instance in $\underline{i}$.

- A *block* is n-variate random variable $(\underline{c}^k(\theta_1, \underline{i}_l), \underline{c}^k(\theta_2, \underline{i}_l), \ldots, \underline{c}^k(\theta_n, \underline{i}_l))$ that corresponds to the computational results on instance $\underline{i}_l$ for each configuration in the race at step k.



- Null hypothesis – all possible rankings of the candidates within each block are equally likely.

F-Race: Algorithm

The optimization problem is tackled by generating a sequence $\Theta_0 = \Theta \supseteq \Theta_1 \supseteq \Theta_2 \supseteq \ldots$

The step from a set Θ_{k-1} to Θ_k is realized as follows

1. At step k, a new instance $\underline{i}_k$ is considered; each candidate $\theta \in \Theta_{k-1}$ still in the race is executed on $\underline{i}_k$ and each observed cost $c(\theta, \underline{i}_k)$ is appended to its $\underline{c}^{k-1}(\theta, \underline{i})$.
2. An aggregate comparison of the arrays $\underline{c}^k(\theta, \underline{i})$ for all $\theta \in \Theta_{k-1}$ is carried out by a statistical test – non-parametric Friedman 2-way analysis of variance by ranks.

   The null hypothesis being that all possible rankings of the candidates within each block are equally likely.
3. If the null hypothesis is rejected, pairwise comparisons between the best candidate and each other one are carried out by means of the t-test. All candidates that result significantly worse than the best one are discarded.

   Otherwise, all candidates in Θ_{k-1} pass to Θ_k.

F-Race: Algorithm

$\Theta^* = \Theta_0$, $ni = 0$
repeat
  randomly choose instance i from set I; run all configurations on Θ^* on i
  $ni = ni + 1$
  if $(ni \geq ni_{min})$
    perform rank-based Friedman test on results for configurations in Θ^*
    on all instances in I evaluated so far
    if (test indicates significant performance differences)
      θ^* = best configuration in Θ^* according to the statistical population parameter
      over instances evaluated so far
        for all $\theta \in \Theta^* \setminus \{\theta^*\}$ **do**
          perform pairwise t-test on θ and θ^*
          if (test indicates significant performance differences)
            eliminate θ from Θ^*
          end if
        end for
    end if
  end if
until (termination condition)
return Θ^*

F-Race: Final Remarks

Good technique, but:

- not suited for applications with large configuration spaces;
- thus, mainly used for configuration problems with few parameters and rather small configuration spaces.