Evaluation Report on Six Experiments

Anne Anonymous

No Institute Given

Abstract. This is the evaluation report on six experiments, namely 1FlipHC, 1FlipHCrs, 2FlipHC, 2FlipHCrs, mFlipHC, and mFlipHCrs on 100 benchmark instances. This report has been generated with the version 0.8.3 of the Evaluator Component of the Optimization Benchmarking Tool Suite.

1 Instance Information

In Figure 2 we illustrate the relative amount of benchmark runs per instance feature. In total, we have 100 benchmark instances and each of them is characterized by two features, namely k and n. The slices in the pie charts are the bigger, the more benchmark instances have the associated feature value, in comparison to the other values. If a slice is bigger than other slices, this therefore means that the used benchmark instances focus on investigating that feature while less runs are applied to the other features.

2 Performance Comparisons

2.1 Estimated Cumulative Distribution Function

We analyze the estimated cumulative distribution function (*ECDF*) [2,3,1] computed based on $\frac{F}{k}$ over $\log_{10} FEs$. The $\text{ECDF}\left(FEs, \frac{F}{k} \leq 0\right)$ represents the fraction of runs which reach a value of $\frac{F}{k}$ less than or equal to 0 for a given ellapsed runtime measured in *FEs*. The *ECDF* is always computed over the runs of an experiment for a given benchmark instance. If runs for multiple instances are available, we aggregate the results by computing their arithmetic mean. The x-axis does not represent the values of *FEs* directly, but instead $\log_{10} FEs$. The *ECDF* is always between 0 and 1 — and the higher it is, the better.

2.2 Estimated Cumulative Distribution Function

We analyze the estimated cumulative distribution function (*ECDF*) [2,3,1] computed based on $\frac{F}{k}$ over $\log_{10} RT$. The $\text{ECDF}\left(RT, \frac{F}{k} \leq 0.01\right)$ represents the fraction of runs which reach a value of $\frac{F}{k}$ less than or equal to 0.01 for a given ellapsed runtime measured in RT. The *ECDF* is always computed over the runs of an