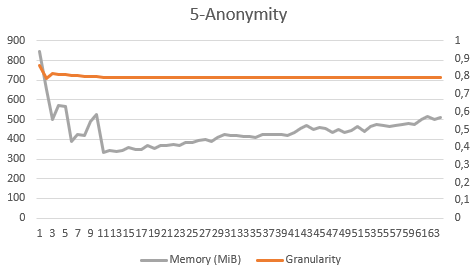
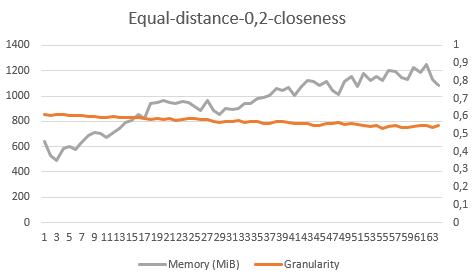
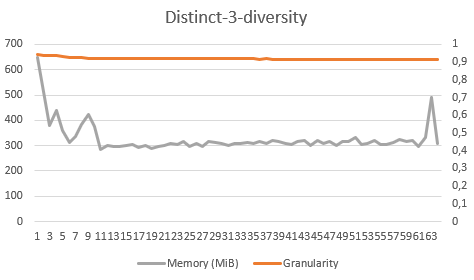
The optimization algorithms supported by ARX use a snapshotting mechanism, where interesting transformations of the dataset with certain properties are cached in memory to speed up analyzing future solution candidates [x]. This means that, as a rule of thumb, memory consumption increases with the execution time of the optimization process (up to a maximum, defined by a user-specified maximum cache size). As another rule of thumb, it can be assumed that the execution time of the optimization algorithm correlates with the “hardness” of the anonymization problem that is being solved.

As the behavior of ARX’s algorithms is easier to analyze in cases where a global transformation model is used, we will use memory consumption in this setting to analyze the memory consumption of the algorithm. As can be seen from the detailed results, the behavior is comparable in the local transformation setup. Figure 1 shows the memory consumption profiles observed for selected privacy models which are representative for all experiments.



Increasing „hardness“ of the problem

**Figure 1**: Three representative memory consumption profiles. The “hardness” of the according anonymization problem increases from left to right.

Intuitively, it would be expected that memory consumption increases linearly with the degree of parallelism, as (in the configuration used in the experiments) each parallel anonymization process can allocate a dedicated cache of the same maximum size independently of the degree of parallelism used.

The profiles obtained for distinct-3-diversity and 5-anonymity represent anonymization problems of low or medium complexity, which is also reflected by the high degree of output data utility that can be obtained. In these cases, as can be seen - and in contrast to what could be expected - the distributed algorithm reduces memory consumption by about 50% with higher degrees of parallelism. This can be explained by the fact that the partitioning of the dataset, also due to the sorting-based partitioning method used, results in anonymization problems that are significantly easier to solve. As a result, the parallel anonymization processes use significantly less of the available cache space. For problems of medium complexity, this significant decrease is followed by a slight linear increase that can be explained by the (expected) increase in the number of caches used.

In contrast, the profile obtained when using 0,2-closeness exhibits the expected linear increase. Here the partitioning of the dataset does not result in sub-problems that are easier to solve than for the complete dataset.