Decomposition is a technique to split a problem in a number of parts such some global property of the problem can be obtained or preserved by independent, distributed, processing of parts of the problem. Decomposition has been used in the database and constraint community to allow distributed parties to find solutions for a set of constraints or to modify a set of constraints independently from each other.

We consider the complexity of finding suitable decompositions in constraint systems distinguishing different decomposition aims.

One such an aim, especially popular in the artificial intelligence community, is finding solutions for constraint systems by distributed computations. Decomposition then should preserve solutions, that is local solutions should be mergeable to a global solution. Another aim, often encountered in the database community, is to be able to preserve consistency while allowing each party to add arbitrary local constraints to its local constraint store. We show that contrary to intuition these two modes do not differ from each other.

Furthermore, we address some computational complexity results w.r.t. finding solutions for constraint systems by applying decomposition techniques.