Excluding Polynomial-time Approximation Schemes for Max Csp

Peter Jonsson[*], Andrei Krokhin[†] and Fredrik Kuivinen[‡]

Abstract. The *maximum constraint satisfaction problem* (Max Csp) is the following computational problem: an instance is a finite collection of constraints on a set of variables, and the goal is to assign values to the variables that maximises the number of satisfied constraints. Max Csp captures many well-known problems (such as Max k-Sat and Max Cut) and is consequently **NP**-hard. Thus, it is natural to study how different restrictions on the allowed constraint types affect the complexity and approximability of Max Csp.

Currently there is essentially only one class of constraint languages which are known to give rise to **NP**-hard Csp problems. We prove that this type of constraint language makes Max Csp hard at gap location 1. A Max Csp problem is said to be *hard at gap location 1* if there exist an $\epsilon < 1$ such that it is **NP**-hard to distinguish instances in which all constraints are simultaneously satisfiable from instances where only an ϵ-fraction of the constraints are simultaneously satisfiable. This notion was introduced by Petrank [Computational Complexity, 4 (1994), pp. 133–157] and is a key concept in our proof. Hardness at gap location 1 rules out the existence of a polynomial-time approximation scheme, unless $\mathbf{P} = \mathbf{NP}$.

We then consider Max Csp restricted to a single constraint type; this class of problems contains, for instance, Max Cut, Max DiCut, and the Max H-Colouring problem. Assuming $\mathbf{P} \neq \mathbf{NP}$, we show that such problems do not admit polynomial-time approximation schemes except in some trivial cases for which they are in **PO**. To prove this result we first reduce the problem to binary relations. The result for binary relations is then obtained by studying the automorphism group of the relation and using the result mentioned above.

Both our hardness results holds even if the number of occurrences of each variable is bounded by a constant. This is in fact also a crucial part of our proof—the inapproximability proofs would not work in their current form if we did not have the bounded occurrence property.

As a byproduct of those results we partially answer two open questions. The first one was posed by Engebretsen et al. [Theor. Comput. Sci., 312 (2004), pp. 17–45] . Given a set of equations over a non-abelian finite group such that all equations are simultaneously satisfiable, Engebretsen et al. asked whether a solution which satisfies a fixed but arbitrarily large fraction of the equations could be found in polynomial time. (Determining if all equations are simultaneously satisfiable is **NP**-hard.) We answer this question negatively, and show that unless $\mathbf{P} = \mathbf{NP}$ such an algorithm do not exist. The second question was posed by Feder et al. [Discrete Math., 307 (2007), pp. 386–392] and concerns the hardness of bounded occurrence Csp(Γ) under the assumption that Csp(Γ) is **NP**-complete. In this case we show that if the current conjecture regarding the complexity of the Csp problem is true, then for every constraint language L such that Csp(L) is **NP**-complete there is an integer k such that Csp(L) with at most k occurrences of each variable is **NP**-complete.

[*]Department of Computer and Information Science, Linköpings Universitet, SE-581 83, Linköping, Sweden (petej@ida.liu.se)

[†]Department of Computer Science, University of Durham, Science Laboratories, South Road, Durham DH1 3LE, UK (andrei.krokhin@durham.ac.uk)

[‡]Department of Computer and Information Science, Linköpings Universitet, SE-581 83, Linköping, Sweden (freku@ida.liu.se)