Structural tractability of enumerating CSP solutions

Gianluigi Greco · Francesco Scarcello

Published online: 1 November 2012
© Springer Science+Business Media New York 2012

Abstract The problem of deciding whether CSP instances admit solutions has been deeply studied in the literature, and several structural tractability results have been derived so far. However, constraint satisfaction comes in practice as a computation problem where the focus is either on finding one solution, or on enumerating all solutions, possibly projected to some given set of output variables. The paper investigates the structural tractability of the problem of enumerating (possibly projected) solutions, where tractability means here computable with polynomial delay (WPD), since in general exponentially many solutions may be computed. A framework based on the notion of tree projection of hypergraphs is considered, which generalizes all structural decomposition methods that are based on decomposing a given instance into suitable tree-like groups of polynomial-time computable subproblems. Tractability results have been obtained both for classes of structures where output variables are part of their specification, and for classes of structures where computability WPD must be ensured for any possible set of output variables. By exhibiting dichotomies, these results are shown to be tight for classes of structures having bounded arity.

Keywords Structural decomposition methods · Enumeration problems · Tree projections · Tree decompositions

G. Greco (✉)
Department of Mathematics, University of Calabria, 87036, Rende (CS), Italy
e-mail: ggreco@mat.unical.it

F. Scarcello
DEIS, University of Calabria, 87036, Rende (CS), Italy
e-mail: scarcello@deis.unical.it

Springer
1 Introduction

1.1 Constraint satisfaction and enumeration problems

Constraint satisfaction is often formalized as a homomorphism problem that takes as its input a pair \((A, B)\) of two finite relational structures \(A\), modeling variables and scopes of the constraints, and \(B\), modeling relations associated with constraints, and asks whether there is a homomorphism from \(A\) to \(B\). Solving this basic decision problem is however not enough in many practical applications. Indeed, one usually needs either to actually compute one solution (i.e., a homomorphism) or, even, to enumerate all possible solutions. In this paper, we deal with the latter case, by formalizing an instance of the constraint satisfaction enumeration problem (short: ECSP) as a triple \((A, B, O)\), whereas \(O\) of “output” variables is given in addition to \(A\) and \(B\), and by looking for algorithms that compute the set of all solutions projected to \(O\).

In fact, this is a natural setting, given that modeling real-world applications typically requires the use of “auxiliary” variables, whose precise values in the solutions are not relevant to the user: In these cases, computing all combinations of their values occurring in solutions would mean wasting time, possibly exponential time.

Since constraint satisfaction is a well-known NP-hard problem, in order to single out “islands of tractability”, many restrictions have been considered in the literature, where the given instances have to satisfy additional conditions. In this paper, we consider restrictions imposed on the so-called left-hand structure \(A\), i.e., \(A\) must be taken from some suitably defined class \(A\) of structures, while the so-called right-hand structure \(B\) can be any arbitrary structure from the class “\(-\)” of all finite structure. Within this setting, the constraint satisfaction problem is often denoted as CSP\((A, -)\).

Analogously to the case of the basic decision problem, restrictions can be applied in the context of the enumeration problem too. For example, the problem of enumerating all homomorphisms (without and without projections) for instances whose right-hand structures are arbitrary ones and whose left-hand structures belong to some suitably defined class \(A\) of structures has been studied in \([4, 6, 10, 40, 41, 54]\). In this paper, we additionally consider restrictions imposed on the set of output variables, and in particular the two possibilities discussed next, both exploited in the literature (see, e.g., \([36]\)).

Output-Aware Instances: In this case, variables that are requested to be output ones are explicitly described in the input structure \(A\). Unlike previous approaches that considered additional “virtual” constraints covering together all possible output variables \([36]\), in this paper possible output variables are described as those variables having a domain constraint, that is, a distinguished unary constraint specifying the domain of this variable. Such variables are said domain restricted. In fact, this choice reflects the classical approach in constraint satisfaction systems, where variables are associated with domains, which are heavily exploited by constraint propagation algorithms. Note that this approach does not limit the number of solutions, while in the tractable classes considered in \([36]\) only instances with a polynomial number of (projected) solutions may be dealt with.

Arbitrary Output Variables: In this case, there is no restriction at all on the allowed output variables. Accordingly, stronger conditions are expected to be needed for tractability. Indeed, for a given structure, the possibility of selecting any set of