Journal of Computer and System Sciences 78 (2012) 681–688

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

The complexity of weighted and unweighted #CSP ☆

Andrei Bulatov [a], Martin Dyer [b,*], Leslie Ann Goldberg [c], Markus Jalsenius [d], Mark Jerrum [e], David Richerby [c]

[a] School of Computing Science, Simon Fraser University, University Drive, Burnaby, Canada, V5A 1S6
[b] School of Computing, University of Leeds, Leeds, LS2 9JT, UK
[c] Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK
[d] Department of Computer Science, University of Bristol, Bristol, BS8 1UB, UK
[e] School of Mathematical Sciences, Queen Mary, University of London, Mile End Road, London, E1 4NS, UK

ARTICLE INFO

Article history:
Received 15 May 2010
Received in revised form 24 October 2011
Accepted 1 December 2011
Available online 8 December 2011

Keywords:
Counting
Constraint satisfaction
Complexity theory

ABSTRACT

We give some reductions among problems in (nonnegative) weighted #CSP which restrict the class of functions that needs to be considered in computational complexity studies. Our reductions can be applied to both exact and approximate computation. In particular, we show that the recent dichotomy for unweighted #CSP can be extended to rational-weighted #CSP.

1. Introduction

The counting complexity of the weighted constraint satisfaction problem, for both exact and approximate computation, has been an active research area for several years. See, for example, [1–18]. The objective is to give a precise categorisation of the computational complexity of problems in a given class. Easily the most significant development in this stream of research was a recent result of Bulatov [1]. This establishes a dichotomy for exact counting in the whole of (unweighted) #CSP. The dichotomy is between problems in FP and problems which are #P-complete. Dyer and Richerby [16] have given an easier proof of this theorem, and have shown it to be decidable [17].

In this paper, we study equivalences among problems in weighted #CSP. These equivalences can greatly simplify the classes of problems which need to be considered in studies of computational complexity. A particular consequence of these results is that the dichotomy for unweighted #CSP can be extended to nonnegative rational-weighted #CSP. In the results we present here, the weights will usually lie in some subset of the nonnegative algebraic numbers, since the proofs do not appear to extend to negative weights [18] or complex weights [5]. Neither do we consider general real numbers, since we want our results to apply to standard models of computation and their complexity classes. An extension to a suitable model of real number computation may be possible, though statements about complexity would need to be modified appropriately.

The plan of the paper is as follows. In Section 1.1 we define the weighted constraint satisfaction problem and establish some notation. In Section 1.2, we define a notion of reducibility, which we call *weighted reduction*, that is used in all our proofs. Its advantage is that the same reductions apply to both exact and approximate computation. Section 2 proves

☆ Research supported by an EPSRC grant "Computational counting" (Dyer, Goldberg, Jerrum and Richerby), and by an NSERC Discovery Grant (Bulatov).
* Corresponding author. Fax: +44 113 343 5468.
E-mail address: M.E.Dyer@leeds.ac.uk (M. Dyer).

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2011.12.002

the equivalence of unweighted and rational-weighted #CSP. Section 3 shows that a weighted #CSP problem can be assumed to have only one function, while retaining several useful restrictions on instances. Finally, in Section 4, we show that any rational-weighted problem is computationally equivalent to an unweighted problem with only binary constraints. Thus any #CSP problem is equivalent to a canonical digraph-labelling problem. This gives another proof of the equivalence of unweighted and rational-weighted #CSP.

1.1. Weighted constraint satisfaction

Let $\mathbb{Z}$, $\mathbb{Q}$, $\overline{\mathbb{Q}}$ and $\mathbb{A}$ denote the integers, rational numbers, real algebraic numbers, and (complex) algebraic numbers, respectively. Let $\mathbb{Z}_\geqslant$, $\mathbb{Q}_\geqslant$ and $\overline{\mathbb{Q}}_\geqslant$ denote the *nonnegative* numbers in $\mathbb{Z}$, $\mathbb{Q}$ and $\overline{\mathbb{Q}}$, respectively. The *positive* integers $\mathbb{Z}_\geqslant \setminus \{0\}$ will be denoted by $\mathbb{N}$, and the positive algebraic numbers $\overline{\mathbb{Q}}_\geqslant \setminus \{0\}$ by $\overline{\mathbb{Q}}_>$. Also $\mathbb{B}$ will denote $\{0,1\}$ and, if $n \in \mathbb{N}$, then $[n]$ will denote $\{1,2,\dots,n\}$.

Let $D = \{0,1,\dots,q-1\}$ ($q \in \mathbb{N}$), which we call the *domain*, and $\mathbb{K} \subseteq \mathbb{A}$, which we call the *codomain*. Let

$$\mathfrak{F}_r(D,\mathbb{K}) = \{f : D^r \to \mathbb{K}\}, \qquad \mathfrak{F}(D,\mathbb{K}) = \bigcup_{r\geqslant 1} \mathfrak{F}_r(D,\mathbb{K}),$$

denote the sets of functions of all *arities* from D to $\mathbb{K}$. We will write $r = r(f)$ for the arity of $f \in \mathfrak{F}(D,\mathbb{K})$. If $r(f) = 1$, f is called a *unary* function and, if $r(f) = 2$, it is a *binary* function.

A problem $\#\mathsf{CSP}(\mathcal{F})$ is parameterised by a finite set $\mathcal{F} \subset \mathfrak{F}(D,\mathbb{K})$ for some D and $\mathbb{K}$. An *instance* I of $\#\mathsf{CSP}(\mathcal{F})$ consists of a finite set of *variables* V and a finite set of *constraints* $\mathcal{C}$. A constraint $\kappa = \langle \mathbf{v}_\kappa, f_\kappa \rangle \in \mathcal{C}$ consists of a function $f_\kappa \in \mathcal{F}$ (of arity $r_\kappa = r(f_\kappa)$) and a *scope*, a sequence $\mathbf{v}_\kappa = (v_{\kappa,1},\dots,v_{\kappa,r_\kappa})$ of variables from V, which need not be distinct. A *configuration* σ for the instance I is a function $\sigma : V \to D$. If $\mathbf{v} = (v_1,\dots,v_r)$, we will write $\sigma(\mathbf{v})$ for $(\sigma(v_1),\dots,\sigma(v_r))$. The *weight* of the configuration σ is given by

$$\mathsf{w}(\sigma) = \prod_{\kappa \in \mathcal{C}} f_\kappa\big(\sigma(\mathbf{v}_\kappa)\big).$$

Finally, the *partition function* $Z_\mathcal{F}(I)$ is given, for an instance I, by

$$Z_\mathcal{F}(I) = \sum_{\sigma : V \to D} \mathsf{w}(\sigma).$$

Then $\#\mathsf{CSP}(\mathcal{F})$ denotes the problem of computing the function $Z_\mathcal{F}$. We will write

$$\#\mathsf{CSP}_q[\mathbb{K}] = \{\#\mathsf{CSP}(\mathcal{F}) \colon \mathcal{F} \subset \mathfrak{F}(D,\mathbb{K}),\ |D| = q\}, \qquad \#\mathsf{CSP}[\mathbb{K}] = \bigcup_{q=2}^{\infty} \#\mathsf{CSP}_q[\mathbb{K}].$$

The case $q = 1$ is clearly trivial, so we omit it from the definition of $\#\mathsf{CSP}[\mathbb{K}]$. The case $q = 2$ is called *Boolean* $\#\mathsf{CSP}[\mathbb{K}]$.

If Γ is a set of *relations*, as in [1,2], we regard it as a set of functions $\mathcal{F}(\Gamma) \subset \mathfrak{F}(D,\mathbb{B})$, so #CSP means $\#\mathsf{CSP}[\mathbb{B}]$. If $R \in \Gamma$ is r-ary, we define $f(R) \in \mathcal{F}$ so that, for each $\mathbf{a} \in D^r$, $f(\mathbf{a}) = 1$ if $\mathbf{a} \in R$, and otherwise $f(\mathbf{a}) = 0$. Then we write $\#\mathsf{CSP}(\Gamma)$ rather than $\#\mathsf{CSP}(\mathcal{F}(\Gamma))$, and Z_Γ rather than $Z_{\mathcal{F}(\Gamma)}$.

We consider here only *non-uniform* #CSP, where D and $\mathcal{F}$ are considered to be objects of constant size. Thus it is only the variable set V, and the constraint set $\mathcal{C}$, that determine the size of an instance.

Various other restrictions on $\#\mathsf{CSP}[\mathbb{K}]$ have been considered in the literature, often in combination. For example, we may insist that $|\mathcal{F}| = m$, for some $m \in \mathbb{N}$, particularly $m = 1$, e.g. [5]. We may insist that no function has arity greater than r, for some $r \in \mathbb{N}$, particularly $r = 2$, e.g. [4]. We may insist that no variable occurs more than k times in an instance, e.g. [10]. We may insist that the functions in $\mathcal{F}$ possess some particular property, such as symmetry, e.g. [13]. We do not consider these restrictions in any detail here. However, we will make use of the following restricted version of $\#\mathsf{CSP}[\mathbb{K}]$ in Section 4.

A unary function which must be applied *exactly once* to each variable $v \in V$ will be called a *vertex weighting*, and its function values *vertex weights*. Thus, if $\lambda : D \to \mathbb{K}$ is a vertex weighting, any instance I must contain exactly one constraint of the form $\langle (v), \lambda \rangle$ for each $v \in V$. Observe that it is not necessary to allow multiple vertex weightings $\lambda_1, \lambda_2, \dots, \lambda_m$, since these can be combined into one equivalent vertex weighting $\lambda = \lambda_1 \lambda_2 \cdots \lambda_m$.

Our definition of vertex weights conforms to the use of similar terminology elsewhere, for example in [15]. We will denote the problem with $\mathcal{F} \subset \mathfrak{F}(D,\mathbb{K})$ and vertex weighting $\lambda : D \to \mathbb{K}$ by $\#\mathsf{CSP}(\mathcal{F};\lambda)$. The problem $\#\mathsf{CSP}(\mathcal{F};\lambda)$ is a restriction on the inputs to an associated $\#\mathsf{CSP}[\mathbb{K}]$ problem, $\#\mathsf{CSP}(\mathcal{F} \cup \{\lambda\})$. In an instance of $\#\mathsf{CSP}(\mathcal{F} \cup \{\lambda\})$, $\langle (v), \lambda \rangle$ can appear any number of times, including zero, for each $v \in V$; in an instance of $\#\mathsf{CSP}(\mathcal{F};\lambda)$, each $\langle (v), \lambda \rangle$ appears precisely once.

We will also consider *approximate* evaluation of $Z_\mathcal{F}$, meaning *relative* approximation. Thus, given $\epsilon > 0$ we wish to compute an estimate $\widehat{Z}_\mathcal{F}(I)$ of $Z_\mathcal{F}(I)$, for all I, such that

$$\big|\widehat{Z}_\mathcal{F}(I) - Z_\mathcal{F}(I)\big| \leqslant \epsilon \big|Z_\mathcal{F}(I)\big|. \tag{1}$$