# On Balanced CSPs with High Treewidth *

**Carlos Ansótegui** and **Ramón Béjar** and **César Fernàndez** and **Carles Mateu**
{carlos,ramon,cesar,carlesm}@diei.udl.cat,
Dept. of Computer Science, Universitat de Lleida, SPAIN

## Abstract

Tractable cases of the binary CSP are mainly divided in two classes: constraint language restrictions and constraint graph restrictions. To better understand and identify the hardest binary CSPs, in this work we propose methods to increase their hardness by increasing the balance of both the constraint language and the constraint graph. The balance of a constraint is increased by maximizing the number of domain elements with the same number of occurrences. The balance of the graph is defined using the classical definition from graph theory. In this sense we present two graph models; a first graph model that increases the balance of a graph maximizing the number of vertices with the same degree, and a second one that additionally increases the girth of the graph, because a high girth implies a high treewidth, an important parameter for binary CSPs hardness. Our results show that our more balanced graph models and constraints result in harder instances when compared to typical random binary CSP instances, by several orders of magnitude. Also we detect, at least for sparse constraint graphs, a higher treewidth for our graph models.

## Introduction

Most of the polynomial time restrictions of CSPs can be divided into two classes, constraint language restrictions (restrictions on the types of constraints that occur) and structural restrictions (constraint hypergraph restrictions). See (Grohe 2006) for a recent survey. Here we focus only on binary CSPs, so we work with constraint (undirected) graphs. To better understand the nature of the hardest CSPs, we propose some methods to increase the hardness of typical instances of random binary CSPs, by modifying the methods for creating the constraints and the constraint graph, by equalizing some of the values that characterize a CSP problem. We consider as our reference model the random binary CSP model B $\langle V, d, C, t\rangle$, extensively studied (see for example (Achlioptas *et al.* 1997; Gent *et al.* 2001)). That is, instances with $V$ variables, domain size $d$, number of constraints $C$ and $t$ valid tuples (or $d^2 - t$ nogoods) per constraint. Previous work has identified some reasons for the hardness of the typical instances on the phase transition (Hogg, Huberman, & Williams 1996; Smith & Dyer 1996; Selman & Kirkpatrick 1996; Prosser 1996) or for the occurrence of *ehps* (exceptionally hard problems) on the underconstrained region (Hogg & Williams 1994; Gent & Walsh 1994; Smith & Grant 1995; 1997; Gomes *et al.* 2004). In this work we focus only on the hardness of the typical instances on the phase transition, so we do not study *ehps* on our CSP models.

In the case of the constraint language, we consider the restriction of balancing the constraints, that is, to balance the number of occurrences of every domain value in the tuples in each side of the constraint. We also consider the additional restriction of symmetry, that can be imposed at the same time of the balance condition. An example of balanced symmetric constraint is the $k-$alldiff binary constraint associated with a graph $k-$coloring problem, that is NP-complete only for $k > 2$. So, not any balanced or symmetric balanced constraint language will be hard, but our empirical results show that they seem to be harder than general random constraint languages.

For getting harder constraint graphs, we consider two models. The first one, in which vertices have at most two different degrees, tries to minimize the probability that any vertex to be more relevant than others when finding a solution. This makes harder for heuristics to choose graph vertices as there are no much differences between them. The second model, where vertices have at most three different degrees, generates graphs with a high girth value, a structural property linked with the graph treewidth. The treewidth has been found to be the most general structural parameter that defines the boundary between polynomial-time and NP-hard restrictions for binary CSPs with arbitrary constraint languages (Grohe 2003). In this paper, we focus on CSPs with sparse, but connected, constraint graphs. With sparse graphs it is more difficult to obtain high treewidth graphs.

We define a balanced CSP as a CSP with a balanced constraint graph and a balanced constraint language. Previous work has used similar ideas to get hard instances for other combinatorial problems (Kautz *et al.* 2001; Ansótegui *et al.* 2006). We present results that indicate that our two graph models generate harder CSP instances than pure random graphs, and that balanced constraints increase even more the hardness of the instances. In addition of explaining the increase of hardness of the constraint graphs due to the balance of the models, we also give results that seem to indicate that our graph models have a higher expected treewidth than pure random graphs, at least in the case of sparse constraint graphs.

*Research partially supported by projects TIN2004-07933-C03-03, TIN2006-15662-C02-02 funded by the *Ministerio de Educación y Ciencia*.

## Balancing the constraint language

### Balanced constraints

The first way to increase the hardness is to balance the number of occurrences of every domain value on each side of the constraint. A constraint $R_{i,j}$ with the same number of occurrences of every domain value on each side of the constraint can be seen as a regular bipartite graph $(V_i \cup V_j, E)$, where part $V_i$ of the graph is associated with side $i$ of the constraint, having an edge $\{a, b\}$ in $E$, with $a \in V_i$ and $b \in V_j$, iff $(a, b) \in R_{i,j}$. That is, the set of allowed tuples $(a, b)$ of the constraint is determined by the set of edges of the regular bipartite graph.

In order to sample from the set of regular bipartite graphs we use a Markov chain algorithm that walks between regular bipartite graphs (Kannan, Tetali, & Vempala 1997). Observe that a perfectly balanced constraint with $t$ tuples must satisfy that $t/d$ is an integer, otherwise we can have an almost balanced constraint. That is, a constraint where $d - (t \ mod \ d)$ domain values appear $\lfloor t/d \rfloor$ times in each side of the constraint and $(t \ mod \ d)$ domain values appear $\lfloor t/d \rfloor + 1$ times in each side of the constraint. This more general constraint is associated with a bi-regular bipartite graph (vertices can have either degree $\lfloor t/d \rfloor$ or $\lfloor t/d \rfloor + 1$), that can be generated with the Markov chain algorithm as well.

### Symmetric balanced constraints

Here we consider an additional restriction over the constraints. In addition to being balanced, we also impose the constraints to be symmetric ($(a, b) \in R_{i,j} \Leftrightarrow (b, a) \in R_{i,j}$) and not to include any tuple of the form $(a, a)$, for any value $a$ of the domain. In this case, a constraint can be associated with a regular (undirected) graph $(V, E)$, where the vertices are the domain values, and $\{a, b\} \in E \Leftrightarrow (a, b) \in R_{i,j} \wedge (b, a) \in R_{i,j}$. It is interesting to consider this additional restriction because it was shown in (Hell & Nešetřil 1990) that a constraint language having only one symmetric constraint (but not necessarily balanced) is NP-hard only when the associated graph is non-bipartite and does not contain loops[1]. Because asymptotically a random graph will not be bi-partite, it is interesting to check empirically whether the additional restriction of symmetric language creates typical instances of equivalent hardness to the previous model, or the frequency of hard instances decreases.

Analogously to the previous model, we need to consider bi-regular graphs, because depending on the number of tuples $t$ not all values can appear the same number of times [2]. To generate the bi-regular graph associated with a symmetric balanced constraint, we use a generalization of the algorithm presented in (Steger & Wormald 1999). In principle, that algorithm is presented only for regular graphs, but it can be easily generalized for the case of graphs with vertices with two different degrees $k$ and $k + 1$. Since the two degrees are contiguous the performance of the generalization of the algorithm seems to be almost identical to the regular case, although in this case we cannot guarantee a perfect uniform distribution of graphs.

[1] Observe that using a bi-partite graph for building a symmetric constraint is not equivalent to what we do in the previous model, because there the bipartite graph is used in a different way.

[2] Also, if $t$ is odd there will be a tuple $(a, b) \in R_{i,j}$ s.t. $(b, a) \notin R_{i,j}$ ($R_{i,j}$ will be almost symmetric).

An interesting question related to balanced, symmetric or not symmetric, constraints is which additional conditions ensure the CSP is NP-complete. For example, the symmetric balanced $k-$alldiff constraint associated with the binary CSP encoding of a graph $k-$coloring problem, is an NP-hard case, but only if $k > 2$.

## Balancing the constraint graph

### Pure random graphs

As far as we know, in typical random distributions of CSPs, the constraint graph has been built following either the $G(V, p)$ or the $G(V, M)$ random graph model. We consider here the model $G(V, M)$, because we are interested in comparing the hardness of this model with others in which also the number of edges is an exact input parameter, and not an expected value as in the $G(V, p)$ model[3].

### Random regular graphs

As the first model for a harder constraint graph we propose regular graphs, i.e. graphs where all the vertices have the same degree. The most obvious reason why these graphs can be harder is that since all the degrees are equal, degree-based search heuristics will be less effective. A less trivial reason can be the following. A graph $G = (V, E)$ is perfectly balanced iff for any subgraph $(V', E')$ of $G$:

$$\frac{|E|}{|V|} \geq \frac{|E'|}{|V'|}$$

That is, the average degree (or edge density), of any subgraph is never greater than in the whole graph $G$. We consider a graph more balanced than other if the number of its subgraphs that satisfy that relation is higher. For graphs where some vertices have a higher degree than others, as in $G(V, p)$ or $G(V, M)$, one may find subgraphs with bigger edge density than the whole graph. However, *any* connected regular graph is perfectly balanced. Balanced graphs will tend to be harder for heuristics, because it will be less easy to isolate potentially overconstrained subproblems during the search, as the edge density of any subproblem will be never higher. Of course, the fact of finding potentially overconstrained subproblems will also depend on the particular constraints between the variables of the subproblem. This is why in this work we have also considered balancing the constraints. Because we want to consider graphs with any desired number of edges, we use the same more general model of bi-regular graphs (and regular graphs when $E * 2/V$ is an integer) that we use for building a symmetric balanced constraint. Not every bi-regular graph will be perfectly balanced, but in general they will be much more balanced than sparse pure random graphs.

As we have said in the introduction, the treewidth of a graph is the most relevant structural parameter concerning the complexity of binary CSPs with arbitrary constraint languages (Grohe 2003). A tree-decomposition of a graph $G = (V, E)$ (Roberston & Seymour 1986) is a pair $(T, \beta)$, where $T$ is a tree $(V^T, E^T)$ and $\beta : V^T \rightarrow 2^V$ such that:

1. For every $v \in V$ the set $\{t \in V^T | v \in \beta(t)\}$ is non-empty and connected in $T$.

[3] For $G(V, p)$ the expected number of edges is $pV(V - 1)/2$ but in $G(V, M)$ the number of edges is always $M$.

2. For every $e \in E$ there is a $t \in V^T$ s.t. $e \subseteq \beta(t)$.

The width of the tree-decomposition is $max\{|\beta(t)| \; |t \in V^T\} - 1$ and the treewidth of $G$ ($tw(G)$) is the minimum width among all tree-decompositions. Concerning the treewidth of random regular graphs, we can use the following lower bound, that is also valid for general graphs, based on the spectrum of the (combinatorial) Laplacian of the graph (Chandran & Subramanian 2003):

$$tw(G) \geq \left\lfloor \frac{3V}{4} \left( \frac{\mu}{\Delta + 2\mu} \right) \right\rfloor - 1$$

Where $\Delta$ is the maximum degree of $G$, the Laplacian of $G$ is the matrix $D - A$, where $D$ is the diagonal matrix in which $D_{i,i}$ is the degree of $v_i$, $A$ is the adjacency matrix of $G$ and $\mu$ is the second smallest eigenvalue of the Laplacian.

For the case of a $k-$regular graph, where $\mu = k - \lambda_2(G)$, this lower bound can be rewritten as:

$$tw(G) \geq \left\lfloor \frac{3V}{4} \left( \frac{k - \lambda_2(G)}{3k - 2\lambda_2(G)} \right) \right\rfloor - 1$$

Where $\lambda_2(G)$ is the second largest eigenvalue of the adjacency matrix of $G$ ($k$ is the largest eigenvalue for a $k-$regular graph). So, the higher the value of $\mu = k - \lambda_2(G)$, also called the spectral gap, the closer the lower bound gets to $V/4$. This shows a connection between expander graphs and graphs with a high value for this treewidth lower bound, because this spectral gap also gives a lower bound on the edge expansion of a graph. Roughly speaking, an expander graph is a graph that, for any, not too big, subset of vertices $S$, its set of neighbors outside $S$ is big, compared with $|S|$. See for example (Sarnak 2004) for formal definitions.

For random $k-$regular graphs, with $k$ fixed, it is known that asymptotically (as $V \rightarrow \infty$) for $\epsilon > 0$ $Probability(\lambda_2(G) \leq 2\sqrt{k-1}+\epsilon) \rightarrow 1$ (Friedman 2004), where $2\sqrt{k-1}$ is, asymptotically, the lowest possible value for $\lambda_2(G)$. Regular graphs with $\lambda_2(G) \leq 2\sqrt{k-1}$ are called Ramanujan graphs. For Ramanujan graphs, we have:

$$tw(G) \geq \left\lfloor \frac{3V}{4} \left( \frac{k - 2\sqrt{k-1}}{3k - 4\sqrt{k-1}} \right) \right\rfloor - 1$$

So, for a fixed and sufficiently high $k$ the lower bound will get very close to $V/4$. As we are considering regular (or bi-regular) graphs with the degrees growing with $V$, we cannot infer directly that asymptotically our graphs will have a spectral gap of order $k - 2\sqrt{k-1}$. Actually, as the degree grows with $V$, the expanding properties of the graphs can be better for higher degrees. For example, some existing constructions of Ramanujan graphs based on certain Cayley graphs are obtained with a degree $\Omega(\sqrt{V})$, whereas similar Cayley graphs with degree $O(\log V)$ are also expander graphs but not as good as Ramanujan graphs (Alon & Roichman 1994). As far as we know the best result for non-constant degree random $k-$regular graphs is, if $k > \sqrt{V \log V}$ then asymptotically $\lambda_2(G) = o(k)$ (Krivelevich *et al.* 2001). So those regular graphs have a high spectral gap, and a high treewidth spectral lower bound.

For the random graph model $G(V,p)$, recent results (Fan Chung & Vu 2004; Coja-Oghlan 2007) indicate that only when $pV >> \log^2 V$, the expected spectral gap tends to be high (although the results do not clarify if as high as with regular graphs). For $pV = O(\log V)$ (when we have the sparsest but still connected random graphs) the results only indicate that the graph will contain a large *subgraph* with large spectral gap. Moreover, most of the existing constructions of good expander graphs (or even Ramanujan graphs) consist of regular graphs of constant degree (See chapter 6 of (Chung 1997) for some examples). By contrast, the results of (Coja-Oghlan 2007) indicate that random graphs $G(V,p)$ with expected constant degree have a spectral gap of $o(1)$, so they are neither expander graphs nor have a high treewidth spectral lower bound. Therefore, theoretical results show a clear difference between random graphs and random regular graphs when considering graphs with expected constant degree, but such a difference is less obvious as we increase the expected degree.

## High girth graphs

Another way to generate graphs with high treewidth is to consider graphs with high expected girth. The girth of a graph is the length of its shortest cycle. In (Chandran & Subramanian 2005) it is shown that if the girth is at least $g$ and the average degree at least $d$, then we have:

$$tw(G) = \Omega \left( \frac{1}{g+1} (d-1)^{\lfloor (g-1)/2 \rfloor} \right)$$

For random $k-$regular graphs it is known that the average girth is only slightly greater than 3 (McKay, Wormald, & Wysocka 2004).

In (Chandran 2003), the authors present an algorithm that, for given $V$ and $k$ (with $k < V/3$), it builds a graph with girth $g$ that satisfies:

$$g \geq \log_k(V) + O(1)$$

Where the vertices of the graph have one of these possible degrees: $k-1$, $k$ or $k+1$. So, the graph is almost regular but may be less balanced than bi-regular graphs. With such a bound on the girth, we have that these graphs satisfy:

$$tw(G) \geq \Omega \left( \frac{(k-1)^{\lfloor (\log_k(V)/2 \rfloor}}{\log_k(V)+1} \right) = \Omega \left( \frac{\sqrt{V}}{\log_k(V)+1} \right)$$

So, we can build graphs, for any value of $V$ and $k < V/3$, and even with $k$ growing with $V$, such that every graph will satisfy that lower bound. Actually, the graphs obtained with this algorithm could have an even higher treewidth[4], because there are graphs with low girth that, however, have a high treewidth, i.e. the complete graph $K_V$ has girth 3 and treewidth $V - 1$.

The algorithm for the generation of such graphs basically proceeds in a greedy fashion, starting with an initially empty graph and adding edges one by one, connecting vertices which are at large distances in the current graph. Here, we use their algorithm with a slight modification. The original algorithm works with $k$ integer, giving a number of edges $\lfloor Vk/2 \rfloor$. So, we cannot use it directly to get graphs with any desired number of edges. However, one can generalize the algorithm to proceed by adding as many edges as desired, instead of stopping when $|E| = \lfloor Vk/2 \rfloor$. Observe that adding more edges will not reduce the treewidth although this can produce a reduction of the girth of the graph.

[4]Our empirical results seem to confirm a higher treewidth.

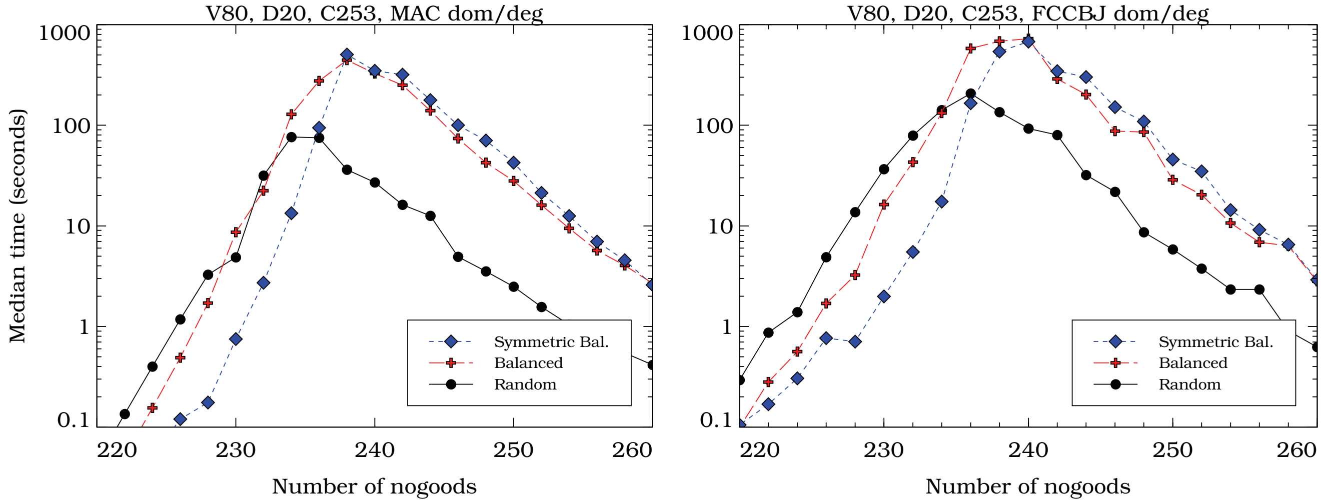


Figure 1: Random constraints versus balanced and symmetric balanced constraints. On top, V stands for number of variables, D for domain size and C for number of constraints.

## Empirical results

All the plots contained in this section are obtained using Forward-Checking with Conflict Back-Jumping (FCCBJ) and Maintaining Arc-Consistency (MAC) search algorithms with variable selection heuristic *dom/deg* (Prosser 1993; Bessière & Régin 1996). Note that other algorithms and heuristics have also been used, for different sizes and constrainedness, showing similar qualitative behavior. More specifically, the same set of problems have been solved with previous mentioned algorithms with *dom+deg* and *dom/deg* heuristics, as well as with Forward Checking (Haralick & Elliott 1980) and a MAC solver using a Satz-like heuristic as described in (Ansótegui *et al.* 2004).

### Balanced constraints

Figure 1 shows the hardness of solving CSPs with either random constraints, balanced constraints or symmetric balanced constraints for sparse constraint graphs ($|edges| = (V/2)\log_2 V$) using MAC and FCCBJ. Differences between random and balanced constraints almost achieve an order of magnitude, meanwhile differences between balanced and symmetric balanced are irrelevant.

### Balanced graphs

Figure 2 shows the hardness of solving CSPs with either pure random, bi-regular or high girth constraint graphs for random constraints. Clearly, our graph models give typical instances of higher complexity than pure random graphs. Between pure random and bi-regular graphs the difference is of several orders of magnitude, and less than 1 order between bi-regular and high girth graphs.

Right plot on Figure 2 shows the hardness for a larger problem solved with the best performing algorithm (MAC). As noted, differences between pure random and our graph models increase with the problem size.

### Bounds on the treewidth

Calculating the treewidth of a graph is NP-hard, and even to approximate it to within a constant absolute additive error (Bodlaender *et al.* 1995), so we cannot get exact results within reasonable time limits. Table 1 shows two different lower bounds and one upper bound on the treewidth of typical instances from the three graph models considered in this paper. The values represent the median of 51 graphs obtained from each model. The first lower bound (Heu LB) is the heuristic lower bound *minimum maximum degree (mmd+)* with *least-c* neighbor selection strategy, as in (Bodlaender, Koster, & Wolle 2004). The second lower bound (Spec LB) is the spectral lower bound explained before. The upper bound (UB) shows the best heuristic upper bound obtained with either *Lexicographic Breadth First Search, variant Minimal (LEX_M)* used in (Koster, Bodlaender, & van Hoesel 2001) or with the min-fill heuristic (Bodlaender 2005)[5].

We observe that as the size increases the spectral lower bound becomes much higher for our more balanced graph models than for pure random graphs. For pure random graphs the heuristic lower bound is the best, but never better than the spectral lower bound for our graph models. So, our graph models seem to have a high spectral gap, and thus a high treewidth lower bound. For the upper bound, we clearly observe that higher values are obtained for our models and that the difference increases with the size. To conclude, even if the gap between the lower and upper bounds is too big to infer exact results about the treewidth, they seem to indicate higher values of the treewidth for our graph models.

### Combined effect

Figure 3 shows the hardness of solving pure random CSPs, CSPs with bi-regular constraint graphs and balanced con-

[5]We thank Arie Koster for providing us with the implementations of the treewidth algorithms and the referees for suggesting to use the min-fill heuristic.

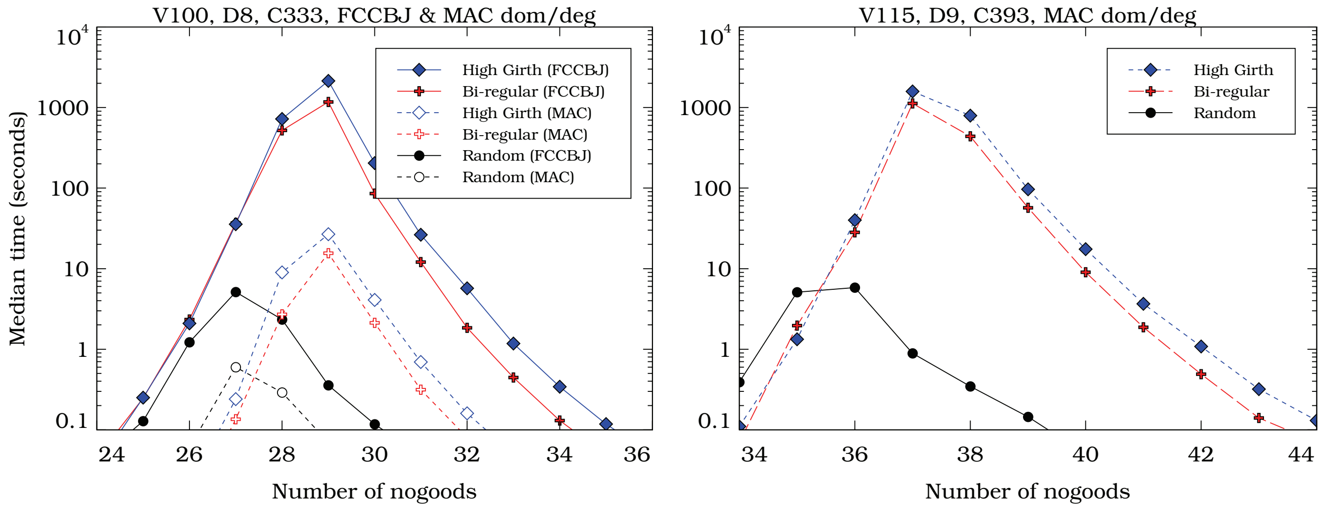


Figure 2: Random graphs versus bi-regular and high girth graphs.

| V | $G(V, M)$ Heu LB | Spec LB | UB | bi-regular Heu LB | Spec LB | UB | high girth Heu LB | Spec LB | UB |
|---|---|---|---|---|---|---|---|---|---|
| 100 | 15 | 3 | 36 | 15 | 13 | 44 | 15 | 14 | 46 |
| 200 | 22 | 6 | 80 | 22 | 28 | 96 | 23 | 30 | 98 |
| 300 | 28 | 14 | 127 | 28 | 43 | 153 | 29 | 45 | 154 |
| 400 | 33 | 12 | 175 | 33 | 61 | 206 | 34 | 63 | 209 |
| 500 | 38 | 22 | 224 | 38 | 79 | 263 | 39 | 81 | 265 |
| 600 | 42 | 28 | 274 | 42 | 92 | 323 | 43 | 94 | 324 |

Table 1: Comparison of lower and upper bounds on the treewidth for the different graphs, with $|edges| = (V/2)\log_2 V$

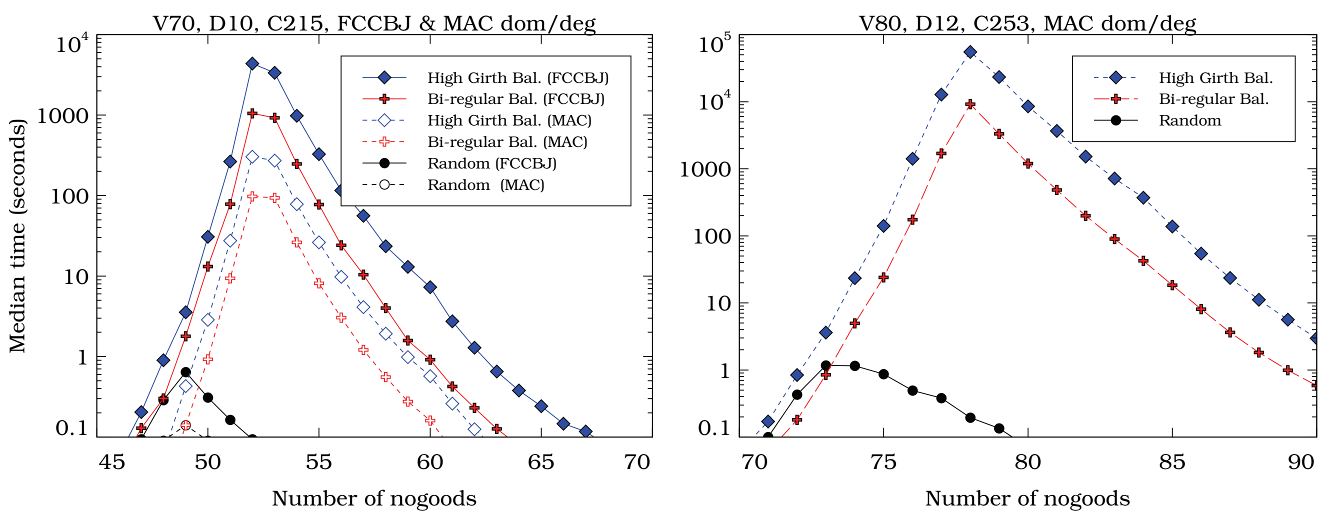


Figure 3: Random CSPs versus balanced CSPs

straints and CSPs with high girth constraint graphs and balanced constraints. This time, the differences between the models are even higher than before: more than 3 orders of magnitude for the best algorithm (MAC) and 4 orders for FCCBJ. The right plot shows that the differences for the best algorithm increase with a larger problem.

So, we observe that the increase in hardness is much more significant when we balance both; the constraint graph and the constraints, and that although the high girth graphs give the hardest instances, more dramatic differences arise between random and bi-regular graphs than between bi-regular and high girth graphs. Our high girth graphs seem to have the highest treewidth, although the bounds are very similar with bi-regular graphs. Also, probably they are not as balanced as bi-regular graphs, although it is not clear that having three contiguous degrees, instead of just two, makes a significant difference on the balance of the graphs. This may partially explain the lower difference between bi-regular and high girth graphs than between bi-regular and random graphs. Probably, high girth graphs as balanced as bi-regular graphs could give an even higher hardness.

## Conclusions

We have defined new random binary CSP models that increase the hardness of the typical instances on the phase transition by increasing the balance on both; the constraint graph and the constraints between the variables. Moreover, our results indicate that another possible explanation for the increased hardness is that our constraint graphs seem to have a higher threewidth than pure random graphs, at least for the sparse constraint graphs we have used.

## References


Achlioptas, D.; Kirousis, L.; Kranakis, E.; Krizanc, D.; Molloy, M.; and Stamatiou, Y. 1997. Random Constraint Satisfaction: a More Accurate Picture. In *Proceedings CP'97*, 107–120.

Alon, N., and Roichman, Y. 1994. Random Cayley graphs and expanders. *Random Structures and Algorithms* 5:271–284.

Ansótegui, C.; del Val, A.; Dotú, I.; Fernández, C.; and Manyà, F. 2004. Modelling choices in quasigroup completion: Sat vs csp. In *Proc. of AAAI-04*.

Ansótegui, C.; Béjar, R.; Fernández, C.; Gomes, C.; and Mateu, C. 2006. The impact of balance in a highly structured problem domain. In *Proc. of AAAI06*, 438–443. AAAI Press.

Bessière, C., and Régin, J.-C. 1996. Mac and combined heuristics: Two reasons to forsake fc (and cbj?) on hard problems. In *CP'96*, 61–75.

Bodlaender, H. L.; Gilbert, J. R.; Hafsteinsson, H.; and Kloks, T. 1995. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *Journal of Algorithms* 18-2:238–255.

Bodlaender, H. L.; Koster, A. M.; and Wolle, T. 2004. Contraction and Treewidth Lower Bounds. *Utrecht University, technical report* UU-CS-2004-034.

Bodlaender, H. L. 2005. Discovering treewidth. In *Proc. of SOFSEM 2005 (LNCS 3381)*, 1–16.

Chandran, L. S., and Subramanian, C. 2003. A spectral lower bound for the treewidth of a graph and its consequences. *Information Processing Letters* 87(4):195–200.

Chandran, L. S., and Subramanian, C. 2005. Girth and treewidth. *Journal of Combinatorial Theory, Series B* 93:23–32.

Chandran, L. S. 2003. A high girth graph construction. *SIAM journal on Discrete Mathematics* 16(3):366–370.

Chung, F. 1997. *Spectral Graph Theory*. Number 92 in Regional Conference Series in Mathematics. AMS.

Coja-Oghlan, A. 2007. On the Laplacian eigenvalues of $G_{n,p}$. *Combinatorics, Probability and Computing*. (to appear).

Fan Chung, L. L., and Vu, V. 2004. The spectra of random graphs with given expected degrees. *Internet Mathematics* 1(3):257–275.

Friedman, J. 2004. A proof of Alon's second eigenvalue conjecture and related problems. *Memoirs of the A.M.S.*

Gent, I., and Walsh, T. 1994. Easy Problems are Sometimes Hard. *AI Journal* 70:335–345.

Gent, I.; MacIntyre, E.; Prosser, P.; Smith, B.; and Walsh, T. 2001. Random constraint satisfaction: flaws and structure. *Constraints* 6(4):345–372.

Gomes, C.; Fernández, C.; Selman, B.; and Bessière, C. 2004. Statistical regimes across constrainedness regions. In *CP'04*.

Grohe, M. 2003. The complexity of homomorphism and constraint satisfaction problems seen from the other side. In *Proc. of FOCS'03*, 552–561.

Grohe, M. 2006. The structure of tractable constraint satisfaction problems. In *Proceedings. MFCS'06*.

Haralick, R., and Elliott, G. 1980. Increasing tree search efficiency for constraint satisfaction problems. *AI Journal* 14:263–313.

Hell, P., and Nešetřil, J. 1990. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B* 48:92–110.

Hogg, T., and Williams, C. 1994. The Hardest Constraint Problems: a Double Phase Transition. *AI Journal* 69:359–377.

Hogg, T.; Huberman, B.; and Williams, C. 1996. Phase Transitions and Search Problems. *Artificial Intelligence* 81 (1-2):1–15.

Kannan, R.; Tetali, P.; and Vempala, S. 1997. Simple Markov-chain algorithms for generating bipartite graphs and tournaments. In *Proc. of the eighth annual ACM-SIAM Symposium on Discrete Algorithms*, 193–200.

Kautz, H.; Ruan, Y.; Achlioptas, D.; Gomes, C.; Bart; ; and Stickel, M. 2001. Balance and filtering in structured satisfiable problems. In *Proc. of IJCAI-01*, 193–200.

Koster, A. M. C. A.; Bodlaender, H. L.; and van Hoesel, C. P. M. 2001. Treewidth: Computational experiments. *Electronic Notes in Discrete Mathematics* 8:54–57.

Krivelevich, M.; Sudakov, B.; Vu, V. H.; and Wormald, N. 2001. Random regular graphs of high degree. *Random Structures and Algorithms* 18:346–363.

McKay, B. D.; Wormald, N. C.; and Wysocka, B. 2004. Short cycles in random regular graphs. *Elect. J. Combinatorics* 11:R66.

Prosser, P. 1993. Domain filtering can degrade intelligent backtracking search. In *Proceedings IJCAI'93*, 262–267.

Prosser, P. 1996. An empirical study of phase transitions in binary constraint satisfaction problems. *AI Journal* 81:81–109.

Roberston, N., and Seymour, P. 1986. Graph minors ii: Algorithmic aspects of treewidth. *Journal of Algorithms* 7:309–322.

Sarnak, P. 2004. What is an expander ? *Notices of the AMS* 51(7).

Selman, B., and Kirkpatrick, S. 1996. Critical behaviour in the computational cost of satisfiability testing. *AI Journal* 81:273–295.

Smith, B., and Dyer, M. 1996. Locating the Phase Transition in Binary Constraint Satisfaction Problems. *AI Journal* 81:155–181.

Smith, B., and Grant, S. 1995. Sparse Constraint Graphs and Exceptionally Hard Problems. In *Proc. IJCAI'95*, 646–651.

Smith, B., and Grant, S. 1997. Modelling Exceptionally Hard Constraint Satisfaction Problems. In *Proc. CP'97*, 182–195.

Steger, A., and Wormald, N. C. 1999. Generating random regular graphs quickly. *Probability and Computing* 8:337–396.