Testing Unsatisfiability of Constraint Satisfaction Problems via Tensor Products

Daya Ram Gaur
Department of Math & Computer Science
University of Lethbridge
Lethbridge, AB, Canada, T1K 3M4
Email: gaur@cs.uleth.ca
http://www.cs.uleth.ca/∼gaur

Muhammad A. Khan
InBridge Inc
3582 30th Street N
Lethbridge, AB, Canada, T1H 6Z4
Email: muhammad@inbridgeinc.com
https://inbridgeinc.com/home-page/team/

Abstract

We study the design of stochastic local search methods to prove unsatisfiability of a constraint satisfaction problem (CSP). For a binary CSP, such methods have been designed using the microstructure of the CSP. Here, we develop a method to decompose the microstructure into graph tensors. We show how to use the tensor decomposition to compute a proof of unsatisfiability efficiently and in parallel. We also offer substantial empirical evidence that our approach improves the praxis. For instance, one decomposition yields proofs of unsatisfiability in half the time without sacrificing the quality. Another decomposition is twenty times faster and effective three-tenths of the times compared to the prior method. Our method is applicable to arbitrary CSPs using the well known dual and hidden variable transformations from an arbitrary CSP to a binary CSP.

Keywords: Constraint Satisfaction, Satisfiability

1 Introduction

Constraint satisfaction problems (CSPs) is a general and descriptive paradigm that is used to model and solve a variety of issues in diverse areas such as scheduling, planning, text analysis and logic. Satisfiability, graph homomorphism, answer set programming are all instances of CSPs. A CSP instance is a set of variables along with a set of relations on subsets of variables. A solution to a CSP satisfies all these relations. CSPs model NP-complete problems and are therefore computationally challenging to solve. One of the practical solution methods is backtracking coupled with constraint propagation. Constraint propagation aims at reducing the search space by enforcing a check on some necessary condition for a solution to exists. One could characterize methods, which use constraint propagation along with backtracking to solve a CSP, as "complete methods" (Ghédira and Dubuisson, 2013). Of course, when a complete method fails to discover a solution to the CSP, we have a "proof" of the unsatisfiability. One can also use an "incomplete method" such as local search to check for unsatisfiability of a CSP at each choice point in the search.

Interest in incomplete methods for detecting unsatisfiability arises from the success of incomplete methods for solving problems; for example, walk-sat for satisfiability problem (Kautz, Sabharwal, and Selman, 2009). A challenge to develop an efficient stochastic local search to determine unsatisfiability for propositional problems was issued by Selman, Kautz, and McAllester (1997). In the same year, Gaur, Jackson, and Havens (1997) gave an incomplete method to determine the unsatisfiability of CSPs. The last twenty years have seen a limited success in addressing the challenge due to Selman et al. (1992) in the context of CSPs. We surmise that there are primarily two reasons for this. The first reason has to do with the size of the microstructure, which is quadratic in the size of the CSP and can result in a computational blow-up. The second reason for limited use of these methods in backtracking is the difficulty in invoking the incomplete method at each choice point. This is due to the non-incremental nature of the technique, that is, one cannot reuse the work done in the prior stages.

In this paper, we exhibit another incomplete method using tensor product of graphs. The key idea is that the microstructure can be described as a union of graph tensor products. We relate the chromatic number of the microstructure to the chromatic numbers of the tensor products in the union. The decomposition of the microstructure into a union of graph tensors implies that upper bounds on the chromatic number of the microstructure can be computed more efficiently in practice. This is due to the facts that the tensor product graphs (i) are smaller than the microstructure, (ii) can be colored in parallel, and (iii) some very efficiently computable upper bounds on the chromatic number are known. Consequently, heuristic algorithms for coloring take significantly lesser time. The known upper bounds on the chromatic number can be used at each choice point. The bookkeeping required for this is extremely fast. Our approach works for arbitrary k-ary CSPs with no restriction on the type of relations. In our approach, coloring a microstructure involves coloring a series of tensor graphs of the same total size as the microstructure. Each of of these computations can be performed in parallel. We give an infinite family of CSPs where the approach can establish unsatisfiability. Although, the use of efficiently computable upper bounds reduces the set of instances where the method can show unsatisfiability, the resulting speed-up is quite significant.

We also report on experiments on random binary CSPs given by the not-all-equal relation. Our experiment design overcomes some of the difficulties (Achlioptas et al., 2001) in the generation of unsolvable instances in (Gaur, Jackson, and Havens, 1997). We observe that a half of all the instances can be proved to be unsatisfiable using the tensor product decomposition for large n (see Figure 6). We also determine unsatisfiability using the original method of Gaur, Jackson, and Havens (1997), which is able to prove unsatisfiability of slightly less than half of the instances (see Figure 6). Our method is almost twice as fast compared to the original method in general. For dense symmetric CSPs our method is twenty times faster, and is able to prove unsatisfiability of three-tenths of the instances compared to the original method (see Figure 4).

In Section 4, we describe a way to decompose the microstructure of a binary CSP into graph tensors. We then relate the chromatic number of the microstructure to the chromatic numbers of constituent tensors. An important property of this decomposition is that the sizes of the individual tensors is smaller than size of the microstructure. Furthermore, the tensors can be colored in parallel and often very efficiently using theoretical results on tensor products and graph coloring. A coloring of the microstructure can be computed very efficiently given a coloring of the tensors. Our approach works for arbitrary CSPs with arbitrary (symmetric or non-symmetric) relations. In Section 5 we give infinite families of CSPs for which unsatisfiability can be detected using the proposed tensor decomposition method. We evaluate our approach empirically and give an efficient method of generating unsatisfiable CSP instances across a spectrum. We observe that the tensor decomposition method proves unsatisfiability of more instances and is twice as fast compared to the original method of Gaur, Jackson, and Havens (1997) on a comprehensive test set. For dense instances it is twenty times faster on average, though the fraction of instances proved unsatisfiable drops down. The tensor decomposition method also improves the incomplete method in (Benhamou and Saïdi, 2008) as discussed in Section 3. Thus our approach partially answers challenge 5 in (Selman, Kautz, and McAllester, 1997). Our tensor decomposition based framework can for unsatisfiability of arbitrary CSPs, not just binary CSPs.

2 Motivation

Constraint solvers have enjoyed resounding success in finding solutions to large scale optimization problems with order of millions of variables. The two issues of 'finding a solution if one exists' and 'showing that there is no solution' are qualitatively very different. The former task is in NP and the latter is in Co-NP. The constraint satisfaction problems generated as shown below do not have a solution. These instances show that the state of art constraint solvers are woefully inadequate at answering the second question. Therefore new algorithms have to designed to show that a CSP does not have a solution.

The instances that we provide come from a conjecture due to Erdős (1981). The conjecture states that any union of n cliques each of order n such that no two cliques intersect in more than one vertex is n-colorable. The conjecture is still open. A natural question to ask is whether the number of cliques can be increased without increasing the chromatic number. In fact there are graphs which are union of $n+1$ cliques each of order n pairwise intersecting in at most one vertex which are not n-colorable.

Let `mat` be a $k+1 \times k$ matrix constructed as follows:

```
var elem = 1;
    for (j =1; j <= k; j++) {
        for (i = 1; i <= j ;i++) {
            mat[i][j] = elem;
            mat[j+1][i] = elem;
            elem = elem +1;
        }
    }
```

The matrix for $n = 8$ is

```
[[1 2 4 7 11 16 22 29]
 [1 3 5 8 12 17 23 30]
 [2 3 6 9 13 18 24 31]
 [4 5 6 10 14 19 25 32]
 [7 8 9 10 15 20 26 33]
 [11 12 13 14 15 21 27 34]
 [16 17 18 19 20 21 28 35]
 [22 23 24 25 26 27 28 36]
 [29 30 31 32 33 34 35 36]]
```

The rows of this matrix are the cliques, and any two cliques intersect in exactly one vertex. The number of cliques is one more than the size of each clique. For each even $n = 4, 6, 8, \dots$ the corresponding graph is not n-colorable. It is worthwhile to note that for odd n, the resulting graph is n colorable. The coloring constraints can be described using pairwise compatibility relations or using n-ary relations. It has been argued that the n-ary allDifferent constraint (Régin, 1994) is a better way to model. An allDifferent constraint on n variables with at most $n-1$ different domain values can be never be satisfied (Hall's Matching Theorem). Therefore, the use of allDifferent constraint in the model easily shows that the graph is not k colorable for all $k \leq n-1$.

We use the allDifferent constraint (one for each clique) to model the coloring problem (with n colors) and solve it using a state of art constraint solver (CP optimizer in IBM ILOG CPLEX 12.9), default options with a parallel search using 32 threads on Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.10GHz. For $n = 4, 6$ the IBM ILOG CPLEX solver took 0.190, 6.570 seconds respectively. For $n = 8$ the program did not finish within 45 minutes.

For $n = 6$, the number of branches was 2,023,823 with 996,880 fails. The search speed (number branches/second) was 876,113.9. The search grows exponentially and it would impossible for the current techniques to prune and show that union of $n+1$ cliques of size n each (as constructed above) is not n-colorable for even $n \geq 8$.

3 Related Research

Several constraint propagation methods: arc-consistency, path-consistency and k-ary consistency (weak and strong) (Montanari, 1974; Mackworth and Freuder, 1985) are in wide spread use. The idea is to infer additional constraints and eventually derive a CSP that is quickly shown to be unsolvable. Régin (1994, 1996) developed constraint propagation methods for all-different constraints. Van Hentenryck (1989) studied bound constraints and cardinality constraints in constraint logic programming.

Various subclasses of CSP are known to be solvable in polynomial time. Freuder (1982) was the first to relate the structure of the CSP to its complexity and showed that a tree structured CSP can be solved in polynomial time using arc-consistency. Dechter and Pearl (1989) introduced the notion of induced-width and related it to the level of consistency required for a backtrack-free search. Gottlob, Greco, and Scarcello (2014) proposed to view the structure of a CSP as a hypergraph, and proved that a CSP with constant hyper-width is solvable in polynomial time. This result subsumes all the above mentioned results relating structure to complexity. For a comprehensive survey on the complexity issues in CSP, see the paper by Carbonnel and Cooper (2016).

Schaefer (1978) dichotomy theorem states that every family of satisfiability instances is either in polynomial time or is NP-complete. Hell and Nešetřil (1990) proved a similar dichotomy theorem for graph homomorphism. Both, satisfiability and graph homomorphism are CSPs. Therefore it is natural to ask the question for CSPs in general. Feder and Vardi (1998) conjectured the existence of a dichotomy theorem for CSPs. Bulatov (2017) recently proved, using algebraic methods, that every family of a non-uniform CSP is either solvable in polynomial time or is NP-complete, thereby establishing the Feder–Vardi conjecture.

The microstructure is a Karp reduction from CSP to k-clique. Therefore, graph-theoretic properties can be used to determine the solvability of a CSP. For instance, if the microstructure is perfect then the CSP can be solved in polynomial time (Salamon and Jeavons, 2008). Recently, a very interesting approach based on patterns, forbidden in the microstructure has been developed. Cooper, Jeavons and Salamon defined the broken triangle property (BTP) in a microstructure with respect to a given variable ordering (Cooper, Jeavons, and Salamon, 2010). They showed that if the microstructure is BTP free with respect to some ordering then the CSP can be solved in polynomial time. Furthermore, the existence of a BTP free ordering can also be determined in polynomial time. Tractability results for several other forbidden patterns have been established by Cohen et al. (2012). A dichotomy is not known for the BTP free property. Hence, several relaxations of the BTP free property have been studied. In particular, m-fBTP (El Mouelhi, 2017, 2018) implies tractability using arc-consistency or arc-consistency with forward checking.

Nearly all of the incomplete methods that test for unsatisfiability, rely on the microstructure (Jégou, 1993). The incomplete method of Gaur, Jackson, and Havens (1997) is to color the microstructure of the CSP using a prescribed number of colors. They showed that the method is intrinsically different from methods based on arc-consistency and also performed a limited empirical evaluation. Bes and Jegou (2005) performed a detailed assessment and argued for the limited applicability of the technique. Benhamou and Saïdi (2008), extended the approach of Gaur, Jackson, and Havens (1997) to propose a new incomplete method based on the notion of dominance in CSPs and established wide applicability. We comment on the relationship of our work to the work of Benhamou and Saïdi (2008). They integrated the approach of Gaur, Jackson, and Havens (1997) with generalized arc consistency (GAC) on all-different constraint due to Régin (1994). Either the original CSP is shown to be unsatisfiable using the method of Gaur, Jackson, and Havens (1997) or a new CSP (a single all-different constraint) is formed using the original CSP and its coloring. GAC on the new CSP may reduce the domains in the original CSP. If that happens, a new microstructure is created and the process iterates. Greedily coloring the microstructure (Step 5, in Algorithm 1) is a crucial component in (Benhamou and Saïdi, 2008). Any improvement to Step 5, improves their algorithm overall. Our decomposition provides a faster way to color the microstructure, thereby improving Step 5 of Algorithm 1 in (Benhamou and Saïdi, 2008).

4 Decomposition into Tensors

We give the definitions and an overview of our approach in this section. The technical details appear in the next Section. A binary CSP is a 3-tuple $(X, D, \mathcal{R})$, where X is the set of variables, D is the set of domain values, and $\mathcal{R}$ is a set of binary relations, called constraints, of pairwise compatible values for pairs of variables in X. Without loss of generality, we assume that each variable $x \in X$ takes values in the domain D. The set of pairwise compatible values for variables x, y are specified by a relation $R_{xy} \subseteq D \times D$. The relation R_{xy} is *symmetric* if $(a, b) \in R_{xy}$ then $(b, a) \in R_{xy}$. Given an assignment $x = a, y = b$, constraint R_{xy} is *satisfied* if $(a, b) \in R_{xy}$. A CSP is said to be *satisfiable* if every variable is assigned a value such that all the constraints are satisfied. If no such assignment of values to the variables exists, then the CSP is *unsatisfiable*. Associated with each CSP is a graph G called the *constraint graph*, in which the nodes are the variables, and (x, y) is an edge if there is a relation $R_{xy} \in \mathcal{R}$.

The following reduction due to (Dechter and Pearl, 1989; Rossi, Petrie, and Dhar, 1990) can be used to transform a non-binary CSP with n constraints into a binary CSP. For each allowed k-tuple of values $(v_1, v_2, \ldots, v_k)$ in a k-ary relation $R(x_1, x_2, \ldots, x_k)$; we have a node (value). An edge connects two k-tuples belonging to different relations R_i, R_j if the "underlying assignment" is compatible. The nodes that belong to the k-tuples from the same constraint form an independent set. Each constraint corresponds to a variable, the k-tuples of values are the values that are allowed by the constraint, in the new binary CSP. The compatible nodes assign the same value to the common variables in the original CSP. A clique in the microstructure of the binary CSP constructed as above corresponds to a solution of the original CSP. If the microstructure can be colored with $< n$ colors, then the non-binary CSP is not satisfiable. There is another reduction method due to Peirce, Hartshorne, and Weiss (1931), which proves that binary CSPs have the same expressive power as arbitrary CSPs. Please see (Rossi, Petrie, and Dhar, 1990) for further details of the reductions. Therefore, from now we will assume without loss of generality that the CSP is binary.

Definition 1 (Microstructure) *Consider a CSP $(X, D, \mathcal{R})$ with constraint graph G and relations in $\mathcal{R}$. The microstructure G_μ of the CSP (Jégou, 1993) is a graph defined as follows: For every unconstrained pair of variables x, y, assume that $R_{xy} \in \mathcal{R}$ is the universal relation allowing all pairs of values for x, y. There is a node x_v for every pair of variable $x \in X$ and value $v \in D$. There is an edge (x_u, y_w) if and only if $x \neq y$ and $(u, w) \in R_{xy}$.*

The following observation can be used to test for the unsatisfiability of a CSP.

Observation 1 *(Gaur, Jackson, and Havens, 1997) An n-variables CSP is unsatisfiable if the microstructure is $(n-1)$-colorable.*

Binary relations can be visualized as directed graphs possibly with loops. If the relation is symmetric we replace the pair of oppositely directed edges by an undirected edge. The not-all-equal relation ($\neq$) over domain $\{1, 2, 3\}$, is the symmetric relation $\neq_{xy} = \{(1, 2), (1, 3), (2, 3), (2, 1), (3, 1), (3, 2)\}$, represented by a triangle graph. The equal ($=$) relation over domain $\{1, 2, 3\}$, is $=_{xy} = \{(1, 1), (2, 2), (3, 3)\}$, represented by a three-node graph with only edges being loops on the nodes. Let C_k be the complete graph with loops on every node. This corresponds to a complete relation on k values (every pair of values is compatible with the relation). Let N_k be the complete graph without any loops. This corresponds to the not-all-equal relation over k values. Given a graph G with k nodes, let G' denote the *complement* of G. Finally, the chromatic number of a graph G denoted $\chi(G)$ is the least number of colors needed to color the nodes of G such that all pairs of nodes that share an edge, have different colors.

Tensor Product of Digraphs

A *digraph* (or directed graph) G with node set (or vertex set) $V(G)$ and edges set $E(G)$ is a graph with each edge endowed with a direction. The (directed) edges of a digraph are typically called *arcs*. Here, we make no notational distinction between an edges joining the nodes a and b and an arc directed from a to b, denoting both by (a, b). The *degree* $\deg_G(v)$ of a node v in a digraph (graph) G is the total number of arcs (edges) meeting the node with loops at v counted twice. We simply write $\deg(v)$ if G is clear from the context.

A digraph G is *symmetric* if $(a, b) \in E(G)$ implies $(b, a) \in E(G)$. A symmetric digraph can be represented by an undirected graph by replacing every pair of symmetric arcs by a single undirected edge. In this way, graphs can be considered special instances of digraphs. The digraphs (graphs) we consider have no parallel arcs (edges), that is, more than one arcs (edges) with the same initial and terminal node (joining the same pair of nodes), but may possess loops. We refer to Chartrand, Lesniak, and Zhang (2015) for the standard terminology concerning

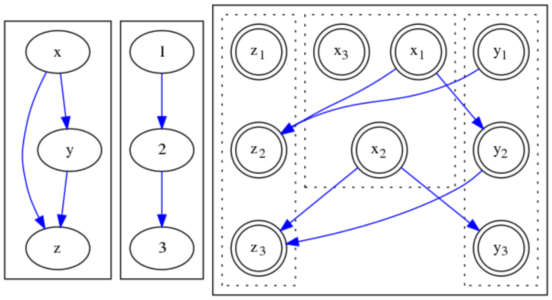


Figure 1: Tensor product ($\otimes$) of digraphs

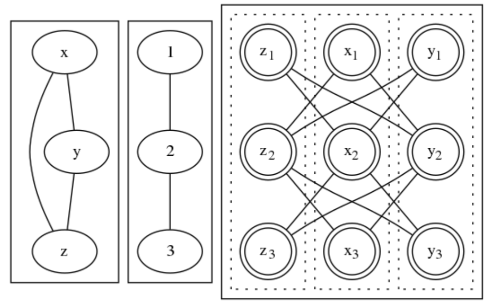


Figure 2: Tensor product ($\otimes$) of graphs

graphs and digraphs. It is easy to observe that any binary relation on a nonempty set can be expressed as a digraph and vice versa. The *underlying graph* of a digraph is the graph obtained by dropping all the arc directions. The *chromatic number of a loopless digraph* is defined to be the chromatic number of its underlying graph.

Introduced as an operation on binary relations in (Whitehead and Russell, 1912), the tensor product can be naturally defined for digraphs (and hence for graphs) as follows (Hammack, Imrich, and Klavzar, 2016). Given digraphs (resp. graphs) G and H, the tensor product $G \otimes H$ has node set $V(G) \times V(H) = \{a_b : a \in V(G), b \in V(H)\}$, the Cartesian product of $V(G)$ and $V(H)$, with an arc directed from a_b to c_d (resp. an edges joining a_b and c_d) if $(a, c) \in E(G)$ and $(b, d) \in E(H)$.

Clearly, $G \otimes H = H \otimes G$. Also, $|V(G \otimes H)| = |V(G)| \cdot |V(H)|$, where $|S|$ denotes the size of a set S. Moreover, for any $a_b \in V(G \otimes H)$, $\deg(a_b) = \deg(a) \cdot \deg(b)$. For any digraphs (or graphs) G and H, $G \otimes H$ contains $|V(H)|$ copies of G and $|V(G)|$ copies of H. Figure 1 shows the tensor product of two digraphs and Figure 2 shows that tensor product of their underlying graphs. We observe that the latter contains twice as many edges as the former. A CSP, along with the decomposition of the microstructure into constituent tensors is shown in Figure 3.

Note that $G \otimes H$ is loopless irrespective of G or H having loops or not, and therefore, $\chi(G \otimes H)$ is well-defined. If G and H are loopless digraphs, then $\chi(G)$ and $\chi(H)$ are also well-defined. Since each copy of G in $G \otimes H$ can be properly colored using $\chi(G)$ colors, we have the following well-known property of tensor products (Hammack, Imrich, and Klavzar, 2016).

Fact 1 *If G and H are loopless digraphs (or graphs), then*

$$\chi(G \otimes H) \leq \min\{\chi(G), \chi(H)\}. \tag{1}$$

When G and H are loopless graphs (Hedetniemi, 1966) conjectured the stronger relation

$$\chi(G \otimes H) = \min\{\chi(G), \chi(H)\}. \tag{2}$$

The digraph analogue of the conjecture does not hold (Poljak and Rödl, 1981). Hedetniemi's conjecture has been one of the most important open problems on graph coloring. It is known to be true in some cases including if one of the graphs is 4-colorable or the complete graph N_k (Hammack, Imrich, and Klavzar, 2016). However, the conjecture has been disproved in general recently (Shitov, 2019).

Decomposition of microstructure

The microstructure graph G_μ of a binary k-CSP $(X, D, \mathcal{R})$ with constraint graph G and $|D| = k$ can also be viewed as a digraph as follows:

Definition 2 (Microstructure - the general case) *The digraph G_μ has a node x_v for every pair of variable $x \in X$ and value $v \in D$. There is an arc (x_u, y_w) if and only if $x \neq y$ and $(u, w) \in R_{xy}$. As before, for every unconstrained pair of variables x, y, we assume that $R_{xy} \in \mathcal{R}$ is the universal relation allowing all pairs of values for x, y.*

Given $x, y \in X$, we denote by G_{xy} the digraph obtained by deleting all arcs from G except (x, y). Recall, that R_{xy} is the relation that lists compatible values for the variables x, y. Comparing the definitions of microstructure and tensor product, we have the following.

Observation 2 $G_\mu = \bigcup_{(x,y) \in E(G)} (G_{xy} \otimes R_{xy}) \bigcup (G' \otimes C_k)$.

If every R_{xy} is the same relation R, one gets the simpler expression

$$G_\mu = (G \otimes R) \cup (G' \otimes C_k).$$

Using the fact that the chromatic number of a union of digraphs is bounded above by the product of their chromatic numbers, we get

$$\chi(G_\mu) \leq \prod_{(x,y) \in E(G)} \chi(G_{xy} \otimes R_{xy}) \cdot \chi(G' \otimes C_k),$$

which reduces to

$$\chi(G_\mu) \leq \chi(G \otimes R) \cdot \chi(G' \otimes C_k),$$

if the relations R_{xy} are all the same.

Now, relations (1) and (2) can be used to upper bound $\chi(G_{xy} \otimes R_{xy})$ or $\chi(G \otimes R)$, but not $\chi(G' \otimes C_k)$ as C_k has loops. However, note that $G' \otimes C_k$ is a graph of significantly smaller number of edges than G_μ. We therefore, have

$$\chi(G_\mu) \leq \prod_{(x,y) \in E(G)} \min\{\chi(G_{xy}), \chi(R_{xy})\} \cdot \chi(G' \otimes C_k),$$

or the reduced form

$$\chi(G_\mu) \leq \min\{\chi(G), \chi(R)\} \cdot \chi(G' \otimes C_k).$$

Since G_{xy}, G, R_{xy} and $G' \otimes C_k$ are all considerably smaller sized graphs than G_μ, we can obtain an upper bound on $\chi(G_\mu)$ rather efficiently.

Loopless k-CSP

If all the relations, considered as digraphs, are loopless in a CSP, then we can expedite the computation of the upper bound. We use the following observation regarding the tensor decomposition and the inequality (1). If a relation (equivalently a digraph) R is partitioned (edges and not the nodes in the digraph are partitioned) into relations R_a and R_b, then for any digraph G,

$$G \otimes R = (G \otimes R_a) \cup (G \otimes R_b),$$

and

$$\chi(G \otimes R) \leq \chi(G \otimes R_a) \cdot \chi(G \otimes R_b).$$

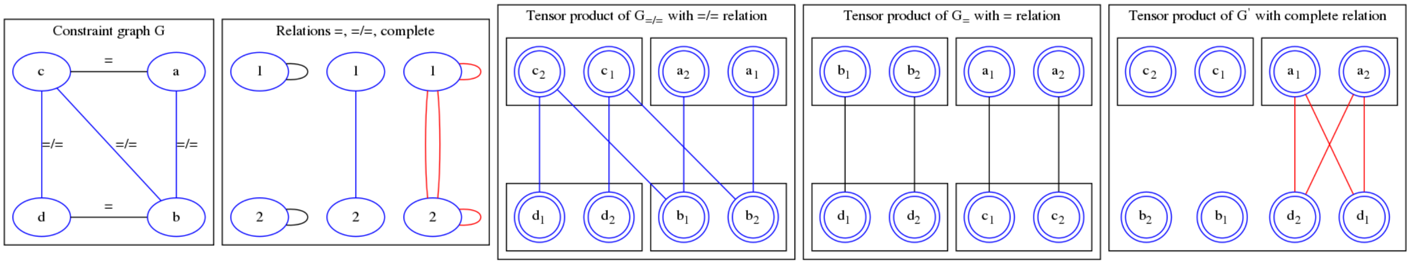


Figure 3: Leftmost graph is the constraint graph G of a CSP with constraints $R_=$, $R_{\neq}$. The subgraph of G in black (blue) is $G_=$ ($G_{\neq}$). The second box contains the three graphs corresponding to relations $=, \neq, C_2$. The third graph is the tensor product $G_{\neq} \otimes R_{\neq}$. The fourth graph is the tensor product $G_= \otimes R_=$. The last graph is the tensor product $G' \otimes C_2$. The labels within () on the nodes correspond to colors.

We partition the complete relation as $C_k = I_k \cup N_k$, where I_k is the '$=$' relation, and N_k is the '$\neq$' relation. For a k-CSP $(X, D, \mathcal{R})$ with all the relations in $\mathcal{R}$ the same loopless relation R, and constraint graph G, the decomposition (1) requires us to compute $\chi(G \otimes R) \cdot \chi(G' \otimes C_k)$, which by the observation above is $\leq \chi(G \otimes R) \cdot \chi(G' \otimes I_k) \cdot \chi(G' \otimes N_k)$. As R and N_k are loopless, we know by (1) that $\chi(G \otimes R) \leq \min\{\chi(G), \chi(R)\} \leq \chi(R) \leq k$. Moreover, since (2) holds when one of the graphs is N_k (complete graph), $\chi(G \otimes N_k) = \min\{\chi(G), \chi(N_k)\} \leq k$. The tensor product $G' \otimes I_k$ is k copies of G', which can be colored by $\chi(G')$ colors. Hence, the following observation.

Observation 3 *If for a loopless k-CSP on n variables, $k^2 \cdot \chi(G') < n$ then the CSP is unsatisfiable. Note that this only requires us to color the complement of the constraint graph resulting in a substantial speed-up.*

As there are infinitely many graphs with a constant chromatic number k such that $k^3 < n$, we have an infinite family for which this decomposition method is able to prove unsatisfiability.

5 Experimental Results

Gaur, Jackson, and Havens (1997) generated random CSPs using the method of Smith (2001). They then identified unsolvable CSPs using an ILP solver. This set is used for testing. Some of the unsolvable instances were arc-consistent, and others were not. The method limits them to small-sized instances ($n = 10$) as the solvability of the CSP needs to be determined. Furthermore, the relation between a pair of variables is asymmetric with a high probability. We conduct our experimental evaluation on symmetric CSPs resulting in symmetric microstructure graphs. The instances are arc-consistent, unsolvable, and of large size. All experiments were implemented on Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz with 8GB of RAM in julia 0.6.2.

We generate an Erdos and Rényi (1960) random graph $G_{n,p}$ over n nodes with an edge probability p. For a constant and fixed p, the family of graphs is known as *dense*. The dense graphs have interesting global properties. For instance, the clique number (the maximum number of pairwise adjacent nodes) of $G_{n,1/2}$ is $2 \log n$, and the maximum degree is $n/2$ almost always, and far away from the chromatic number of $G_{n,1/2}$, which is close to $n/(2 \log n)$ almost always (Bollobás, 1988). Thus, for any $k < n/2 \log n$, the $G_{n,1/2}$ is not k-colorable. This gives us an easy way to obtain unsolvable symmetric k-CSPs, avoiding the use of an ILP solver as in (Gaur, Jackson, and Havens, 1997). Recall that a CSP is arc-consistent if, for any pair of variables (x, y), for all values in the domain of x, there is some consistent value in the domain of y (and vice-versa). The resulting instances are therefore arc-consistent for all $k \geq 2$.

We do not use the method known as Model B (Smith, 2001) for generation of random CSPs. Model B was studied in (Achlioptas et al., 2001). They showed that the model is flawed is the sense that the random CSP instances do not have asymptotic threshold and almost all instances they are over-constrained. Unsatisfiability of these can be readily determined using trivial local inconsistencies. We wanted to avoid unsatisfiability detection by an easy use of local consistency checking. Therefore we used Erdos-Reyni random graphs to highlight cases where local consistency fails, but our method succeeds in practice.

We consider the loopless symmetric k-CSP corresponding to the '$\neq$' relation with constraint graph $G_{n,p}$. The microstructure $G_\mu = G_{n,p} \otimes N_k \cup G'_{n,p} \otimes C_k$ is the union of the two tensor products. The unsatisfiability of such a CSP can be proved in three ways.

The pseudo-code for the three methods below is listed in the appendix.

- μ (microstructure) method: This is the original method of Gaur, Jackson, and Havens (1997). We color the microstructure G_μ and check if $\chi(G_\mu) < n$. If so, the CSP is unsolvable.
- $\otimes$ (tensor decomposition) method: This is one of the decomposition methods developed in this paper. If $\chi(G_{n,p} \otimes N_k) \cdot \chi(G'_{n,p} \otimes C_k) < n$ then the CSP $G_{n,p}$ is unsolvable. Since, equation (2) is satisfied when one of the graphs is N_k, we have $\chi(G_{n,p} \otimes N_k) = k$. To determine the unsatisfiability using the second method we only color $\chi(G'_{n,p} \otimes C_k)$.
- $\otimes_E$ (fast tensor decomposition) method: Observation 3 gives a third way. We obtain a coloring of the complement of the constraint graph and if $k^2\chi(G') < n$, then the CSP is unsatisfiable.

We use the following greedy algorithm as a baseline to color a graph. We color a node using the first available color, given an ordering of the nodes. If no color is available, then we increase the size of the palette. We consider six orderings; the nodes are ordered in the decreasing order of degree, and five random ones. One can replace this step with a method that chooses the next node to color dynamically, such as Brelaz's method. This replacement can only improve the results.

We generated $G_{n,p}$ instances for n in the interval $[10, 100]$ in steps of 10 and p in $[0.1, 1]$ in steps of 0.1. We used the above three methods to color the microstructure. For each p, 1620 instances were generated. The number of instances in the increasing order of n are $300, 600, 900, 1200, 1500, 1800, 2100, 2400, 2700, 2700$. The number of instances in the increasing order of $k \geq 2$ are $3000, 2700, 2400, 2100, 1800, 1500, 1200, 900, 600$.

We first present the aggregate results. Next, we examine the effect of varying the parameters. Finally, we describe the running times. Table 1 lists the number of instances proved unsat-

μ	$\otimes$	$\otimes_E$	#
false	false	false	6744
true	false	false	144
true	true	false	5614
true	true	true	3216
false	true	false	482

Table 1: Number of Unsolvable instances $G_{n,p}$

isfiable by a subset of methods. A total of 9456 instances of 16200 were proved unsatisfiable using one of the methods. The μ, $\otimes$, $\otimes_E$ methods proved unsatisfiability of 8974,9312,3216 instances respectively. The decomposition based methods proved unsatisfiability of 482 additional instances which were not detected by the old method. The old method detected 144 additional instances than the other methods.

The number of instances proved unsolvable by the three methods when grouped by n, p, k are shown in the Table 2. 'X' indicates missing values. For each grouping by n, p, k we determine the fraction of the total instances that are proved unsolvable by the first two strategies ($\mu, \otimes$). The three plots (Figures 4, 5, 6) give the relative performance of the two strategies. The fraction of the instances determined unsolvable by coloring the microstructure are represented on the x-axis, whereas the y-axis represents the fraction of the instances with a proof of unsatisfiability obtained by coloring the tensor product. Figure 7 show the relative performance of the faster tensor decomposition based strategy $\otimes_E$ compared to the $\otimes$ strategy grouped by n, p, k. The percentage of instances proved unsatisfiable by $\otimes$ ($\otimes_E$) strategies are on the x-axis (y-axis).

Figure 4: Relative Performance w.r.t. p

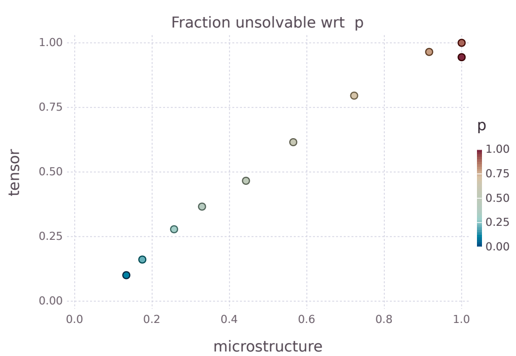
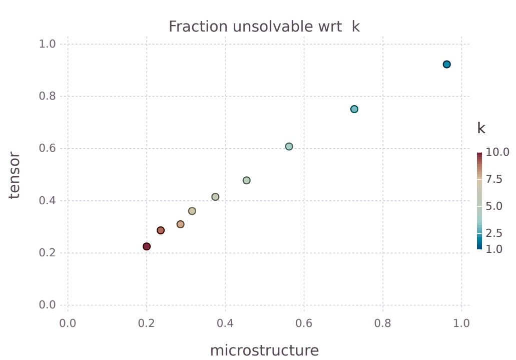


Figure 5: Relative Performance w.r.t. k



Finally, a comment on the running times. If both $G \otimes N_k$ and $G' \otimes C_k$ are to be colored, then they can be colored in parallel, thereby reducing the time. The earlier method (Gaur, Jackson, and Havens, 1997) is inherently sequential. The average times

Figure 6: Relative Performance w.r.t. n

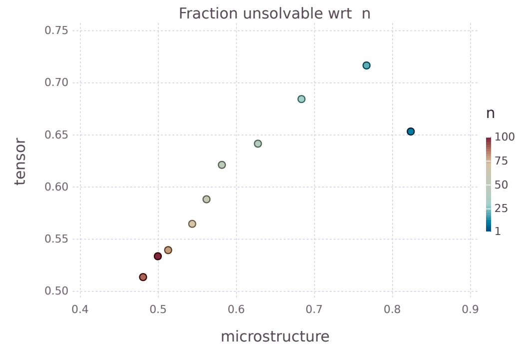
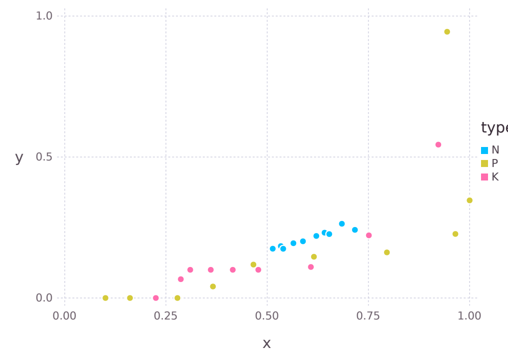
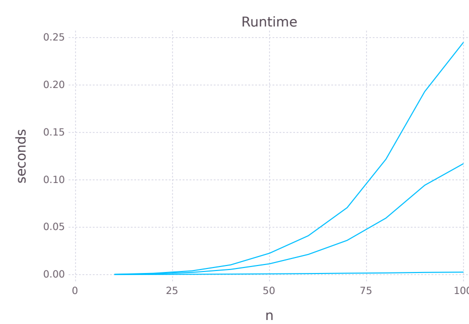


Figure 7: Relative Performance of $x = \otimes$ and $y = \otimes_E$



as a function of n is shown in Figure 8. The curves in the order from the highest to the lowest are for μ, $\otimes$, $\otimes_E$ respectively. The proposed method $\otimes$ is twice as a fast as the old method μ. The faster tensor method $\otimes_E$ is faster by a factor of 40 for values of $p > 0.75$. The $\otimes_E$ method is about 20 times faster on average (over all values of p), and can prove unsatisfiability of 20% of all the instances considered.

Figure 8: Runtime



6 Discussion

Let us call a CSP, a 2-CSP if all the variable domains are of size 2. In our experiments, we observe that the relative fraction of provably unsolvable 2-CSP instances is high (> 0.55) for all the methods. A coloring 2-CSP can be decided in polynomial time. We suspect, the randomized coloring method proves unsatisfiability, almost always for 2-CSPs. A proof of this would be serious progress towards answering challenge 5 in (Selman, Kautz, and McAllester, 1997).

The tensor decomposition methods are applicable for any CSP with an arbitrary number of different k-ary relations on pairs of variables. In our experiments, we assumed that all the binary relations are the same and symmetric (not-all-equal). It

n	μ	$\otimes$	$\otimes_E$	p	μ	$\otimes$	$\otimes_E$	k	μ	$\otimes$	$\otimes_E$
10	247	196	68	0.1	216	163	X	X	X	X	X
20	460	430	145	0.2	283	261	X	2	2887	2768	1632
30	615	616	237	0.3	416	451	X	3	1964	2028	600
40	753	770	278	0.4	533	593	66	4	1348	1459	264
50	872	932	330	0.5	717	755	192	5	953	1004	210
60	1011	1059	362	0.6	915	997	237	6	674	747	180
70	1141	1186	408	0.7	1170	1289	262	7	473	541	150
80	1230	1295	419	0.8	1484	1563	368	8	343	372	120
90	1297	1387	472	0.9	1620	1620	561	9	212	258	60
100	1348	1441	497	1.0	1620	1530	1530	10	120	135	X

Table 2: Number of provably unsatisfiable instances grouped by n, k, p. Columns μ ($\otimes$)[$\otimes_E$] is the number of instances proved unsolvable by coloring the microstructure (tensor product of G' with complete relation C_k)[coloring G']

would be interesting to conduct further experiments in the generalized setting. If there are r different relations, possibly asymmetric, then there are $r+1$ tensor products that have to be colored. They can be colored in parallel, reducing the running time by a factor of $O(1/(r+1))$. The upper bound on the chromatic number of the microstructure is computed as a product of the chromatic numbers of the $r+1$ tensor products. A theoretical analysis, establishing that $r = 1$ is the worst-case (or prove otherwise) for the tensor decomposition method is the second interesting question. Some immediate progress can be made here by assuming that $m < r$ relations give rise to loopless digraphs, in which case we can use Observation 3, and bound the chromatic number by the number of nodes in the component tensor.

Third, and very interesting question is whether the product upper bound on the chromatic number in the decomposition can be strengthened. This requires the development of new heuristic ways of combining the colorings of the tensor products. Any progress on this question would immediately increase the efficacy of our method. This in turn would further speedup the method of Benhamou and Saïdi (2008). We end with a note. In our illustrations the tensor decomposition is based on the relations in the CSP. However, this does not have to be the case, the decomposition is not fixed. In fact, a single CSP relation can be decomposed into multiple relations, (and multiple CSP relations can be combined into one). This raises the possibility of the development of other general methods for decomposing the microstructure graphs into their constituent tensors.

Acknowledgements: The authors thank the referees for the comments.

7 Appendix

This section lists the pseudo-code for the three methods $\mu, \otimes, \otimes_E$. We assume the existence of a method `color(G)`, which colors a graph G greedily, and returns the number of colors used. We also assume existence of the tensor product operator ($\otimes$). julia language provides the `kron` operator which can be used to construct the adjacency matrix of $G \otimes R$, given the adjacency matrices for G, R.

We use the following convention in the listings below. G is the constraint graph. $R_1, R_2, \ldots, R_m$ are the relation graphs on the edges of G. $G(R_i)$ is the graph obtained using the subset of edges in G constrained by relation R_i. The variables have the same domain D and the number of values in D is k.

Algorithm 1 is the method of Gaur, Jackson, and Havens (1997). The pseudo-code in Algorithm 2 describes the tensor decomposition based method. We illustrate the parallelism while coloring the tensors in the decomposition using the `@parallel` construct in julia. This executes the for loop in parallel and the results of individual computations are combined (reduced) using the * operator.

Algorithm 1 μ method (Gaur, Jackson, and Havens, 1997)

Require: Binary CSP, given by G and $R_i's$.
procedure $\mu(G, R_1, R_2, \ldots, R_m)$
  $G_\mu = \phi$ ▷ Initialize G_μ to an empty graph.
  for $i \in [1..m]$ **do** ▷ Compute the microstructure
    $G_\mu = G_\mu \cup (G(R_i) \otimes R_i)$
  $G_\mu = G_\mu \cup (G' \otimes C_k)$
  return `color`(G_μ) ▷ Color the microstructure

Algorithm 2 $\otimes$ method

Require: Binary CSP, given by G and $R_i's$.
procedure $\otimes(G, R_1, R_2, \ldots, R_m)$
  colors = 1
  colors = `@parallel (*)` ▷ Execute the for loop in parallel and reduce the results using the (*) operator
  for $i \in [1..m]$ **do**
    if R_i is loopless **then**
      return $\min\{|G(R_i)|, |R_i|\}$ ▷ Using Eq (1)
    else
      return `color`$(G \otimes R_i)$
  return colors * `color`$(G' \otimes C_k)$

Finally, we describe the method based on Observation 3. This method works only when all the relations are loopless. It colors just the complement of the constraint graph. The number of colors needed for the tensors in the decomposition are estimated using (1).

Algorithm 3 $\otimes_E$ method

Require: Binary CSP, given by G and $R_i's$. $R_i's$ are loopless.
procedure $\otimes(G, R_1, R_2, \ldots, R_m)$
  colors = 1
  for $i \in [1..m]$ **do**
    colors = colors * $\min\{|G(R_i)|, |D|\}$ ▷ O(1) computation.
  ▷ Decompose C_k as $I_k \cup N_k$
  return colors * $|D|$ * `color`(G')

References

Achlioptas, D.; Molloy, M. S.; Kirousis, L. M.; Stamatiou, Y. C.; Kranakis, E.; and Krizanc, D. 2001. Random constraint satisfaction: A more accurate picture. *Constraints* 6(4):329–344.

Benhamou, B., and Saïdi, M. R. 2008. A new incomplete method for CSP inconsistency checking. In *AAAI*, 229–234.

Bes, J.-N., and Jegou, P. 2005. Proving graph un-colorability with a consistency check of CSP. In *Tools with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on*, 2–pp. IEEE.

Bollobás, B. 1988. The chromatic number of random graphs. *Combinatorica* 8(1):49–55.

Bulatov, A. A. 2017. A dichotomy theorem for nonuniform CSPs. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, 319–330. IEEE.

Carbonnel, C., and Cooper, M. C. 2016. Tractability in constraint satisfaction problems: a survey. *Constraints* 21(2):115–144.

Chartrand, G.; Lesniak, L.; and Zhang, P. 2015. *Graphs and Digraphs*. Chapman and Hall/CRC.

Cohen, D. A.; Cooper, M. C.; Creed, P.; Marx, D.; and Salamon, A. Z. 2012. The tractability of CSP classes defined by forbidden patterns. *Journal of Artificial Intelligence Research* 45:47–78.

Cooper, M. C.; Jeavons, P. G.; and Salamon, A. Z. 2010. Generalizing constraint satisfaction on trees: Hybrid tractability and variable elimination. *Artificial Intelligence* 174(9-10):570–584.

Dechter, R., and Pearl, J. 1989. Tree clustering for constraint networks. *Artificial Intelligence* 38(3):353–366.

El Mouelhi, A. 2017. A BTP-based family of variable elimination rules for binary csps. In *AAAI*, 3871–3877.

El Mouelhi, A. 2018. On a new extension of BTP for binary CSPs. *Constraints* 1–28.

Erdős, P. 1981. On the combinatorial problems which i would most like to see solved. *Combinatorica* 1:25–42.

Erdos, P., and Rényi, A. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* 5(1):17–60.

Feder, T., and Vardi, M. Y. 1998. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing* 28(1):57–104.

Freuder, E. C. 1982. A sufficient condition for backtrack-free search. *Journal of the ACM (JACM)* 29(1):24–32.

Gaur, D. R.; Jackson, W. K.; and Havens, W. S. 1997. Detecting unsatisfiable CSPs by coloring the micro-structure. In *AAAI/IAAI*, 215.

Ghédira, K., and Dubuisson, B. 2013. *Constraint Satisfaction Problems*. John Wiley. chapter CSP solving algorithms, 7398.

Gottlob, G.; Greco, G.; and Scarcello, F. 2014. Treewidth and hypertree width. *Tractability: Practical Approaches to Hard Problems* 1.

Hammack, R.; Imrich, W.; and Klavzar, S. 2016. *Handbook of Product Graphs*. CRC Press.

Hedetniemi, S. 1966. Homomorphisms of graphs and automata. Technical report, University of Michigan.

Hell, P., and Nešetřil, J. 1990. On the complexity of h-coloring. *Journal of Combinatorial Theory, Series B* 48(1):92–110.

Jégou, P. 1993. Decomposition of domains based on the micro-structure of finite constraint-satisfaction problems. In *AAAI*, volume 93, 731–736.

Kautz, H. A.; Sabharwal, A.; and Selman, B. 2009. *Handbook of Satisfiability*. IOS Press. chapter Incomplete algorithms, 185203.

Mackworth, A. K., and Freuder, E. C. 1985. The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artificial intelligence* 25(1):65–74.

Montanari, U. 1974. Networks of constraints: Fundamental properties and applications to picture processing. *Information sciences* 7:95–132.

Peirce, C. S.; Hartshorne, C.; and Weiss, P. 1931. *Collected papers of Charles Sanders Peirce. Vol. 3*. Belknap Press of Harvard University Press.

Poljak, S., and Rödl, V. 1981. On the arc-chromatic number of a digraph. *J. Combin. Theory Ser. B* 31(2):190–198.

Régin, J.-C. 1994. A filtering algorithm for constraints of difference in CSPs. In *AAAI*, volume 94, 362–367.

Régin, J.-C. 1996. Generalized arc consistency for global cardinality constraint. In *Proceedings of the thirteenth national conference on Artificial intelligence-Volume 1*, 209–215. AAAI Press.

Rossi, F.; Petrie, C. J.; and Dhar, V. 1990. On the equivalence of constraint satisfaction problems. In *ECAI*, volume 90, 550–556.

Salamon, A. Z., and Jeavons, P. G. 2008. Perfect constraints are tractable. In *International Conference on Principles and Practice of Constraint Programming*, 524–528. Springer.

Schaefer, T. J. 1978. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, 216–226. ACM.

Selman, B.; Levesque, H. J.; Mitchell, D. G.; et al. 1992. A new method for solving hard satisfiability problems. In *AAAI*, volume 92, 440–446. Citeseer.

Selman, B.; Kautz, H.; and McAllester, D. 1997. Computational challenges in propositional reasoning and search. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, 50–54.

Shitov, Y. 2019. Counterexamples to Hedetniemi's conjecture. *arXiv:1905.02167 [math.CO]* 1–3.

Smith, B. M. 2001. Constructing an asymptotic phase transition in random binary constraint satisfaction problems. *Theoretical Computer Science* 265(1-2):265–283.

Van Hentenryck, P. 1989. Constraint satisfaction in logic programming.

Whitehead, A. N., and Russell, B. 1912. *Principia Mathematica*, volume 2. Cambridge University Press.