Perfect Constraints Are Tractable

András Z. Salamon[1,2] and Peter G. Jeavons[1]

[1] Computing Laboratory, University of Oxford, UK
{Andras.Salamon,Peter.Jeavons}@comlab.ox.ac.uk
[2] The Oxford-Man Institute for Quantitative Finance
Andras.Salamon@oxford-man.ox.ac.uk

Abstract. By using recent results from graph theory, including the Strong Perfect Graph Theorem, we obtain a unifying framework for a number of tractable classes of constraint problems. These include problems with chordal microstructure; problems with chordal microstructure complement; problems with tree structure; and the "all-different" constraint. In each of these cases we show that the associated microstructure of the problem is a perfect graph, and hence they are all part of the same larger family of tractable problems.

1 Introduction

Considerable effort has been devoted to identifying tractable subclasses of the constraint satisfaction problem. Most of this work has focused on just *two* general approaches: either identifying forms of constraint which are sufficiently restrictive to ensure tractability no matter how they are combined, or else identifying structural properties of constraint networks which ensure tractability no matter what forms of constraint are imposed (see Chapters 7 and 8 of [10]).

However, some important tractable classes of problems do not fall into either of these categories. A notable example is the class of problems where all the variables must be assigned different values. The tractability of this problem has been exploited to great effect by designing an efficient propagator for the global "all-different" constraint [9], which is widely used in practical constraint solvers.

The binary disequality relation used to specify the "all-different" problem is *not* contained in any tractable language (as it can express the NP-complete Graph Colouring problem). The structure of the "all-different" problem is also *not* a tractable structure (since each variable constrains each other variable, and an arbitrary binary constraint problem can be represented on such a complete structure). It is the *combination* of structure and constraint language that leads to tractability for this problem, and there is currently little theory available to analyse such "hybrid" reasons for tractability.

In this paper we analyse such hybrid properties by examining the properties of a graph associated with every constraint problem instance, known as the *microstructure*. We show that for "all-different" problems, as well as several other known tractable classes, this microstructure is a *perfect* graph (see Figure 1 for a summary). For all such problems the tractability can then be deduced as an immediate consequence of classical results about perfect graphs [5].

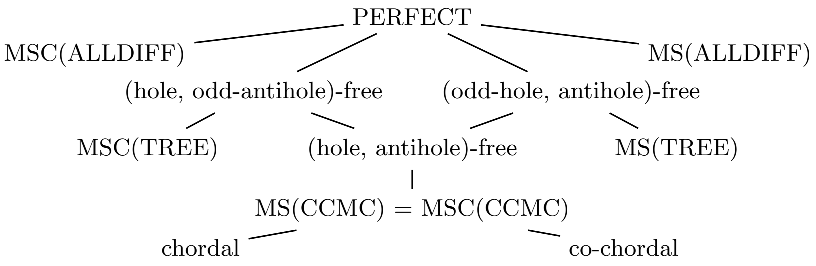

Fig. 1. Inclusions among constraint problems with perfect microstructure.

2 Graphs, Perfect Graphs, and Microstructures

A ***graph*** is a structure $G = (V(G), E(G))$ containing a set $V(G)$ of vertices and a set $E(G) \subseteq \{\{u, v\} \mid u, v \in V(G), u \neq v\}$ of edges. The ***order*** of a graph is the number of its vertices. The ***complement*** $\overline{G}$ of graph G contains the same vertices as G, and its edges are the non-edges of G.

A graph G is a ***subgraph*** of H, written $G \subseteq H$, if $V(G) \subseteq V(H)$ and $E(G) \subseteq E(H)$. The graph H ***contains*** G if G is a subgraph of H such that $E(G)$ contains all edges in $E(H)$ that have both endpoints in $V(G)$. In this case G is also known as an ***induced subgraph*** of H. If G does *not* contain a graph that is isomorphic to any graph in class $\mathcal{C}$, then we say that G is ***$\mathcal{C}$-free***.

A ***clique*** is a subgraph which has edges connecting each pair of vertices. A ***cycle*** of order k in a graph is a subgraph with vertices $\{v_1, \ldots, v_k\}$ and edges $\{v_k, v_1\}$ and $\{v_i, v_{i+1}\}$ for $i = 1, 2, \ldots, k-1$. Such a cycle is usually denoted C_k.

A ***vertex-colouring*** of a graph is an assignment of colours to the vertices of the graph, such that the vertices of every edge are assigned different colours. The ***chromatic number*** of a graph is the smallest number of colours required for a vertex-colouring. A graph G is ***perfect*** if for every induced subgraph H of G, the chromatic number of H is equal to the order of the largest clique contained in H [5, 6]. The smallest non-perfect graph is C_5 (which has chromatic number 3, but maximum clique size 2). Perfect graphs are important for our purposes because of the following classical result.

Theorem 1 ([5, Sect. 6]). *A maximum clique in a perfect graph can be found in polynomial time.*

A ***hole*** is a cycle of order $n \geq 5$. An ***antihole*** is the complement of a hole. The results in this paper rely heavily on the following recent result.

Theorem 2 ([2, 1.2], Strong Perfect Graph Theorem). *A graph is perfect if and only if it is (odd-hole,odd-antihole)-free.*

It has also recently been established that perfect graphs can be recognised in polynomial-time [1]. The class of perfect graphs will be denoted PERFECT.

The ***microstructure*** (MS) of a constraint problem instance is a graph where the set of vertices corresponds to the set of possible assignments of values to variables: a vertex (x, a) represents the assignment of value a to variable x [7].

The edges of the microstructure connect all pairs of variable-value assignments that are allowed (simultaneously) by the constraints. Note that if there is no explicit constraint between two variables x and y, then the microstructure includes edges between all pairs of variable-value assignments involving x and y.

The ***microstructure complement*** (MSC) is the complement of the microstructure: its edges represent pairs of variable-value assignments that are *disallowed* by the constraints [3] (including distinct values for the same variable).

We denote the microstructure and microstructure complement of a constraint problem instance P by MS(P) and MSC(P), respectively, and let MS($\mathcal{C}$) and MSC($\mathcal{C}$) denote the classes of graphs formed by microstructures and microstructure complements of instances in class $\mathcal{C}$. A ***perfect constraint problem*** is a class $\mathcal{C}$ of binary constraint problem instances such that MS($\mathcal{C}$) is perfect.

Theorem 3. *Any perfect constraint problem can be solved in polynomial-time.*

Proof. For binary constraint problems, a solution corresponds to a (maximum) clique in the microstructure of order n, where n is the number of variables [7]. The result follows by Theorem 1. ☐

3 Examples of Perfect Constraint Problems

A graph is called ***chordal*** or *triangulated* if it is (C_{n+4})-free, where C_{n+4} denotes all cycles of order at least 4. Such graphs may have a cycle of order 4 or greater as a subgraph, but not as an *induced* subgraph — in other words, every cycle of order at least 4 must have a "chord" (an edge connecting two of its vertices that are not adjacent in the cycle). A graph is called ***co-chordal*** if its complement is chordal. All chordal and co-chordal graphs are perfect [6].

Jégou noted that binary constraint problems with chordal MS form a tractable class [7]. Cohen noted that binary constraint problems with chordal MSC also form a tractable class [3]. In fact this latter class has been shown to consist of problems that are "permutably max-closed" [4]. We observe that both of these classes are perfect, and can be combined to obtain a larger tractable class. Let CCMC be the class of binary constraint problem instances P, such that either MS(P) is chordal or MSC(P) is chordal.

Proposition 4. *MS(CCMC) = MSC(CCMC)* $\subset$ *(hole,antihole)-free*[3]

Proof. The antihole of order 5 is isomorphic to C_5 and all larger antiholes contain C_4, so chordal graphs are antihole free. The remaining inclusions are easy. ☐

It is well-known that tree-structured binary constraint problems are tractable [10, Chapter 7]. However, the MS of a tree-structured problem is no longer a tree, and nor is the MSC. For example, the MSC of a single binary disequality constraint contains C_4, so it is not a tree (and is also not chordal). On the other hand, tree-structured problems are perfect, as we now show. Let TREE be the class of all tree-structured binary constraint problem instances.

---

[3] Also called *weakly chordal.*

Proposition 5. *MSC(TREE) $\subset$ (hole,odd-antihole)-free $\subset$ PERFECT.*

Proof. Let $G = \text{MSC}(P) \in \text{MSC(TREE)}$. If G contains a cycle C_k of order $k \geq 5$, then the vertices of C_k must involve at least 3 different variables (since different values for the same variable are all connected in a microstructure complement), and this implies that the structure of the instance P must contain a cycle, which contradicts the fact that it is tree-structured. Hence G is hole-free.

Since the antihole of order 5 is isomorphic to C_5 we need only consider antiholes of order ≥ 7. Assume for contradiction that G contains an antihole A of order $k \geq 7$, on the vertices $(v_0, v_1, \ldots, v_{k-1})$ (in that order around the cycle). We note that A contains an induced subgraph isomorphic to C_4 on every 4 vertices $(v_i, v_{i+1}, v_{j+1}, v_j)$ such that $0 \leq i < j < k$ and $i < i+2 < j < j+2 < i+k$ (with subscripts taken modulo k). Any induced 4-cycle in the MSC of a tree-structured problem must involve exactly 2 variables, hence every set of 4 vertices of this kind in A involves exactly 2 variables. This implies that the vertices $(v_1, v_2, \ldots, v_k)$ involve just two variables, which alternate around the cycle.

If k is odd, these conditions are unsatisfiable, so G is odd-antihole-free. □

We also obtain the following symmetrical result on taking complements.

Corollary 6. *MS(TREE) $\subset$ (odd-hole,antihole)-free $\subset$ PERFECT.*

An "***all-different***" constraint can be represented by a set of binary constraints of the form $x \neq y$, for each pair of distinct variables x and y.

The microstructure complement of such a constraint problem contains edges between vertices (x, a) and (y, b) (corresponding to assignments $x = a$ and $y = b$ respectively), when either $x = y$ and $a \neq b$, or else $a = b$ and $x \neq y$. For example, the MSC of an "all-different" problem with 3 variables, with domains $\{a_1, a_2\}, \{a_2, a_3\}, \{a_1, a_3\}$, is the even hole, C_6. Hence "all-different" problems do not lie in any of the tractable classes described so far.

Let ALLDIFF be the class of "all-different" constraint problem instances. A ***gridline*** graph is one whose vertices can be embedded in the real plane, such that there is an edge between two distinct vertices precisely when they are on the same horizontal line or the same vertical line [8].

Proposition 7. *MSC(ALLDIFF) = gridline $\subset$ (odd-hole,odd-antihole)-free.*

Proof. Let $G = \text{MSC}(P) \in \text{MSC(ALLDIFF)}$. We can embed each vertex (x, a) of G in the real plane, using the associated variable, x, to determine the horizontal position, and the associated value, a, to determine the vertical position. Now the edges of G are precisely those required by the definition of gridline.

Conversely, for any gridline graph G we have an associated embedding in the real plane, so we can map each vertex to a pair (x, y), and consider these to be the (variable,value) pairs of a constraint problem. The graph G can then be considered as the MSC of an all-different problem.

Hence MSC(ALLDIFF) = gridline. By the results of [8], gridline graphs are (odd-hole,odd-antihole)-free (and hence perfect, by Theorem 2). □

We also obtain the following symmetrical result on taking complements.

Corollary 8. *MS(ALLDIFF) $\subset$ (odd-hole,odd-antihole)-free = PERFECT.*

4 Conclusions and Future Work

We have shown that a wide range of constraint problem classes (including hybrid classes which have previously been difficult to analyse) have a perfect microstructure, and hence can be solved efficiently by a single method.

However, the classical algorithm for finding a maximum clique in a perfect graph [5], although polynomial, is currently too slow to be practically useful, although there has been some progress in improving the method [11]. Recent advances such as the Strong Perfect Graph Theorem could provide a new route to developing more practical algorithms. The results presented above provide a strong additional motivation to develop such algorithms, which could then be used to solve a wide variety of constraint problems in a unified way.

Conversely, it may be that some of the algorithmic ideas developed for constraint solvers and propagators could be used in a more general, graph-theoretic context. For example, for gridline graphs, Peterson suggests that matching techniques could be used to obtain $O(n^3)$ algorithms for finding a maximum independent set [8], and that better algorithms should exist. The algorithm developed by Régin for obtaining domain consistency in all-different constraints is based on matching in a bipartite graph [9], which can be achieved in time $O(n^{2.5})$, and it may be that similar ideas could be exploited for arbitrary gridline graphs.

References

1. Chudnovsky, M., Cornuéjols, G., Liu, X., Seymour, P., Vušković, K.: Recognizing Berge graphs. Combinatorica **25**(2) (2005) 143–186
2. Chudnovsky, M., Robertson, N., Seymour, P., Thomas, R.: The strong perfect graph theorem. Annals of Mathematics **164**(1) (2006) 51–229
3. Cohen, D.A.: A new class of binary CSPs for which arc-consistency is a decision procedure. In: Proceedings of CP'03. Number 2833 in LNCS (2003) 807–811
4. Green, M.J., Cohen, D.A.: Domain permutation reduction for constraint satisfaction problems. Artificial Intelligence **172**(8–9) (2008) 1094–1118
5. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica **1**(2) (1981) 169–197
6. Hougardy, S.: Classes of perfect graphs. Discrete Math. **306** (2006) 2529–2571
7. Jégou, P.: Decomposition of domains based on the micro-structure of finite constraint-satisfaction problems. In: Proceedings AAAI'93. (1993) 731–736
8. Peterson, D.: Gridline graphs: a review in two dimensions and an extension to higher dimensions. Discrete Applied Mathematics **126** (2003) 223–239
9. Régin, J.C.: A filtering algorithm for constraints of difference in CSPs. In: Proceedings AAAI'94. (1994) 362–367
10. Rossi, F., van Beek, P., Walsh, T., eds.: Handbook of Constraint Programming. Elsevier (2006)
11. Yıldırım, E.A., Fan-Orzechowski, X.: On extracting maximum stable sets in perfect graphs using Lovász's theta function. Comp. Optim. and Appl. **33** (2006) 229–247