A Characterisation of the Complexity of Forbidding Subproblems in Binary Max-CSP*

Martin C. Cooper[1], Guillaume Escamocher[1], and Stanislav Živný[2]

[1] IRIT, University of Toulouse III, 31062 Toulouse, France
cooper@irit.fr, escamocher@irit.fr
[2] Department of Computer Science, University of Oxford, OX1 3QD Oxford, UK
standa.zivny@cs.ox.ac.uk

Abstract. Tractable classes of binary CSP and binary Max-CSP have recently been discovered by studying classes of instances defined by excluding subproblems. In this paper we characterise the complexity of all classes of binary Max-CSP instances defined by forbidding a single subproblem. In the resulting dichotomy, the only non-trivial tractable class defined by a forbidden subproblem corresponds to the set of instances satisfying the so-called joint-winner property.

1 Introduction

Max-CSP is a generic combinatorial optimization problem which consists in finding an assignment to the variables which satisfies the maximum number of a set of constraints. Max-CSP is NP-hard, but much research effort has been devoted to the identification of classes of instances that can be solved in polynomial time.

One classic approach consists in identifying tractable constraint languages, i.e. restrictions on the constraint relations which imply tractability. For example, if all constraints are supermodular, then Max-CSP is solvable in polynomial time, since the maximization of a supermodular function (or equivalently the minimization of a submodular function) is a well-known tractable problem in Operations Research [18]. Over two-element domains [8], three-element domains [14], and fixed-valued languages [11], a dichotomy has been given: supermodularity is the only basic reason for tractability. However, over four-element domains it has been shown that other tractable constraint languages exist [15]. Another classic approach consists in identifying structural reasons for tractability, i.e. restrictions on the graph of constraint scopes (known as the constraint graph) which imply the existence of a polynomial-time algorithm. In the case of binary CSP the only class of constraint graphs which ensures tractability (subject to certain complexity theory assumptions) are essentially graphs of bounded tree-width [9, 12]. It is well known that structural reasons for tractability generalise to optimisation versions of the CSP [1, 10].

* Martin Cooper is supported by Projects ANR-10-BLAN-0210 and 0214. Stanislav Živný is supported by a Junior Research Fellowship at University College, Oxford.

Recently, a new avenue of research has led to the discovery of tractable classes of CSP or Max-CSP instances defined by forbidding a specific (set of) subproblem(s). Novel tractable classes have been discovered by forbidding simple 3-variable subproblems [4, 7]. In the present paper we consider all classes of binary Max-CSP instances defined by forbidding occurrences of a single subproblem. The dichotomy that we give can be seen as an important first step towards a complete characterisation of the complexity of classes of binary Max-CSP instances defined by forbidding *sets* of subproblems.

To relate this to similar work on the characterisation of the complexity of forbidden patterns [3], we should point out that a pattern can represent a set of subproblems by leaving the compatibility of some pairs of assignments undefined. Another difference between subproblems and patterns is that in a subproblem all variable-value assignments are assumed distinct, whereas in a pattern two assignments may represent the same assignment in an instance [3]. In a related avenue of research, other workers have defined tractable classes of binary CSP instances by excluding (sets of) induced subgraphs in the microstructure of the instance, where the microstructure of a CSP instance is the graph $\langle V, E \rangle$ with V the set of all variable-value assignments and $\{p, q\} \in E$ if and only if the pair of variable-value assignments p, q are compatible [13, 2, 19]. These microstructure-based tractable classes of CSP instances do not generalise to tractable classes of Max-CSP instances. Indeed, we are not aware of any tractable classes of Max-CSP defined exclusively in terms of the microstructure.

The complexity of classes of binary Max-CSP instances defined by local properties of (in)compatibilities have previously been characterised, but only for properties on exactly 3 assignments to 3 distinct variables [5]. In the present paper we consider classes defined by forbidding subproblems of any size and possibly involving several assignments to the same variable, thus allowing more refinement in the definition of classes of Max-CSP instances.

2 Definitions and Basic Results

A *subproblem* P is simply a binary Max-CSP instance: variables are distinct, each variable has its own domain composed of distinct values, and a cost of 0 or 1 is associated with each pair of assignments to two distinct variables. In a Max-CSP instance defined in this way, the goal of maximising the number of satisfied constraints is clearly equivalent to minimising the total cost. We consider that in any subproblem or instance a constraint is given for each pair of distinct variables (even if the constraint corresponds to a constant-0 cost function). We only consider subproblems with all binary constraints but no unary constraints. As we will show later, our results are independent of the presence of unary constraints. It will sometimes be more convenient to consider an instance as a set of variable-value assignments together with a function *cost*, such that $cost(p, q) \in \{0, 1\}$ denotes the cost of simultaneously making the pair of assignments p, q, together with a function *var* such that $var(p)$ indicates the variable associated with assignment p.

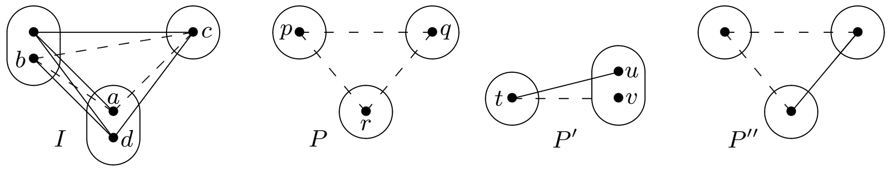

Fig. 1. The instance I contains P and P' as subproblems but not P''.

A subproblem P *occurs* in a binary Max-CSP instance (or, equivalently, another subproblem) I if P is isomorphic to some sub-instance of I obtained by taking a subset U of the variables of I and subsets of each of the domains of the variables in U. We also say that I *contains* P as a subproblem. To illustrate this notion, consider the instance I and the three subproblems P, P', P'' shown in Fig. 1. A bullet point represents a variable-value assignment, assignments to the same variable are grouped together in the same oval, a dashed line between points a and b means $cost(a, b) = 1$ and a solid line means $cost(a, b) = 0$. In this example, subproblem P occurs in I with the corresponding isomorphism $p \mapsto a$, $q \mapsto b$, $r \mapsto c$. Similarly, P' occurs in I with the corresponding isomorphism $t \mapsto c$, $u \mapsto d$, $v \mapsto a$. On the other hand, P'' does not occur in I.

In this paper we denote by $\mathcal{F}(P)$ the set of Max-CSP instances in which the subproblem P is forbidden, i.e. does not occur. Thus if I, P' and P'' are as shown in Fig. 1, $I \in \mathcal{F}(P'')$ but $I \notin \mathcal{F}(P')$. If $\Sigma = \{P_1, \ldots, P_s\}$ is a set of subproblems, then we use $\mathcal{F}(\Sigma)$ or $\mathcal{F}(P_1, \ldots, P_s)$ to denote the set of Max-CSP instances in which no subproblem $P_i \in \Sigma$ occurs. The following lemma follows from the above definitions, by transitivity of the occurrence relation.

Lemma 1. *If* $\forall P \in \Sigma_1$, $\exists Q \in \Sigma_2$ *such that* Q *occurs in* P, *then* $\mathcal{F}(\Sigma_2) \subseteq \mathcal{F}(\Sigma_1)$.

We say that $\mathcal{F}(\Sigma)$ is *tractable* if there is a polynomial-time algorithm to solve it. We say that $\mathcal{F}(\Sigma)$ is *intractable* if it is NP-hard. We assume throughout this paper that $P \neq NP$. Suppose that $\mathcal{F}(\Sigma_1) \subseteq \mathcal{F}(\Sigma_2)$. Clearly, $\mathcal{F}(\Sigma_1)$ is tractable if $\mathcal{F}(\Sigma_2)$ is tractable and $\mathcal{F}(\Sigma_2)$ is intractable if $\mathcal{F}(\Sigma_1)$ is intractable. Our aim is to characterise the tractability of $\mathcal{F}(P)$ for all subproblems P. We first show that we only need to consider subproblems with domains of size at most 2.

Lemma 2. *Let* P *be a subproblem with three or more values in the domain of some variable and let* $\mathcal{F}(P)$ *be the set of Max-CSP instances in which the subproblem* P *is forbidden. Then* $\mathcal{F}(P)$ *is intractable.*

Proof. Max-Cut is intractable and can be reduced to Max-CSP on Boolean domains [8]. Thus $\mathcal{F}(P)$ is intractable since it includes all instances of Max-CSP on Boolean domains. □

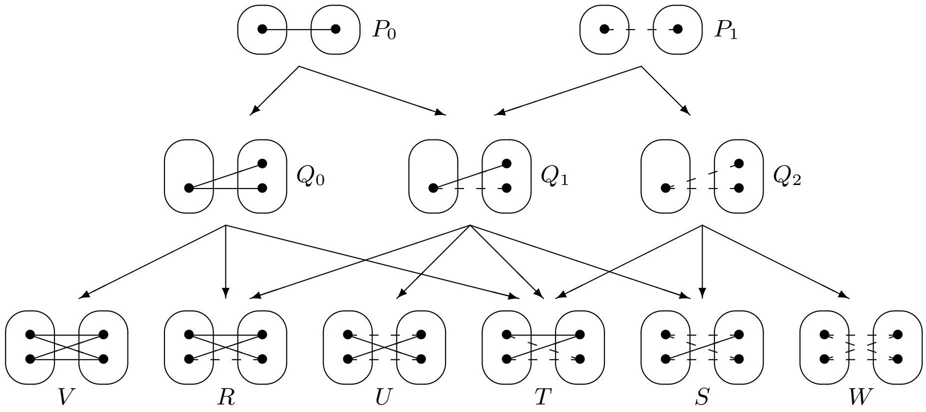

Fig. 2. Subproblems on two variables (showing inclusions between subproblems).

3 Subproblems on Two Variables

We now consider the subproblems on just two variables shown in Fig. 2. Modulo independent permutations of the variables and of the two domains, these are the only possible subproblems with domains of size at most 2.

Lemma 3. *If Q_1 is the subproblem shown in Fig. 2, then $\mathcal{F}(Q_1)$ is intractable.*

Proof. Let I be an instance in $\mathcal{F}(Q_1)$. It is easy to see that all binary cost functions between any pair of variables in I must be constant. Hence I is equivalent to a trivial Max-CSP instance with no binary cost functions. □

Lemma 4. *If Q_0 and U are as shown in Fig. 2, then $\mathcal{F}(\{Q_0, U\})$ is intractable.*

Proof. Max-Cut can be coded as Max-CSP over Boolean domains in which all constraints are of the form $X_i \neq X_j$. We can replace each constraint $X_i \neq X_j$ by an equivalent gadget G with two extra variables Y_{ij}, Z_{ij}, where G is given by $\neg X_i \wedge Y_{ij}$, $\neg Y_{ij} \wedge \neg X_j$, $X_i \wedge \neg Z_{ij}$, $Z_{ij} \wedge X_j$. It is easily verified that placing the gadget G on variables X_i, X_j is equivalent to imposing the constraint $X_i \neq X_j$; when $X_i = X_j$ at most one of these constraints can be satisfied and when $X_i \neq X_j$ at most two constraints can be satisfied.

For each pair of variables X, X' in the resulting instance of Max-CSP such that there is no constraint between X and X', we place a binary constraint on X, X' of constant cost 1. In the resulting Max-CSP instance, there are no two zero costs in the same binary cost function. Thus, this polynomial reduction from Max-Cut produces an instance in $\mathcal{F}(\{Q_0, U\})$. Intractability of $\mathcal{F}(\{Q_0, U\})$ then follows from the NP-hardness of Max-Cut. □

Lemma 5. *If Q_2 and U are as shown in Fig. 2, then $\mathcal{F}(\{Q_2, U\})$ is intractable.*

Proof. As in the proof of Lemma 4, the proof is again by a polynomial reduction from Max-Cut. This time each constraint $X_i \neq X_j$ is replaced by the gadget G' where G' is $\neg X_i \vee Y_{ij}$, $\neg Y_{ij} \vee \neg X_j$, $X_i \vee \neg Z_{ij}$, $Z_{ij} \vee X_j$. When $X_i = X_j$ at most three of these constraints can be satisfied and when $X_i \neq X_j$ all four constraints can be satisfied.

For each pair of variables X, X' in the resulting instance of Max-CSP such that there is no constraint between X and X', we place a binary constraint on X, X' of constant cost 0. The resulting instance is in $\mathcal{F}(\{Q_2, U\})$. □

This provides us with a dichotomy for subproblems on just two variables.

Theorem 1. *If P is a 2-variable binary Max-CSP subproblem, then $\mathcal{F}(P)$ is tractable if and only if P occurs in Q_1 (shown in Fig. 2).*

Proof. By Lemma 2, we only need to consider subproblems in which each domain is of size at most two.

Since each of P_0 and P_1 occur in Q_1, it follows from Lemma 3 and Lemma 1 that $\mathcal{F}(P_0)$ and $\mathcal{F}(P_1)$ are also tractable. Since Q_0 occurs in R, T, V and Q_2 occurs in S, W, it follows from Lemmas 4, 5 and Lemma 1 that $\mathcal{F}(Q_0)$, $\mathcal{F}(Q_2)$, $\mathcal{F}(R)$, $\mathcal{F}(S)$, $\mathcal{F}(T)$, $\mathcal{F}(U)$, $\mathcal{F}(V)$, $\mathcal{F}(W)$ are all intractable. This covers all the possible subproblems with domains of size at most 2 as shown in Fig. 2. □

4 Subproblems on Three Variables

We recall the following result which follows directly from Theorem 5 of [5].

Lemma 6. *A class of binary Max-CSP instances defined by forbidding a single subproblem comprised of a triangle of three assignments to three distinct variables is tractable if and only if the three binary costs are 0,1,1.*

Binary Max-CSP instances in which the triple of binary costs 0,1,1 does not occur in any triangle satisfy the so-called joint-winner property [7]. This class has recently been generalised to the hierarchically-nested convex class which is a tractable class of Valued CSP instances involving cost functions of arbitrary arity [6]. The following corollary is just a translation of Lemma 6 into the notation of forbidden subproblems.

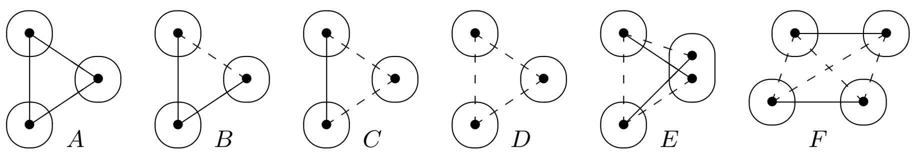

Fig. 3. Subproblems on three or four variables

Corollary 1. *Let A, B, C, D be the subproblems shown in Fig. 3. Then $\mathcal{F}(C)$ is tractable, but $\mathcal{F}(A)$, $\mathcal{F}(B)$, and $\mathcal{F}(D)$ are intractable.*

Lemma 7. *Given the subproblem E shown in Fig. 3 and the set $\mathcal{F}(E)$ of Max-CSP instances in which the subproblem E is forbidden, then $\mathcal{F}(E)$ is intractable.*

Proof. The constraint graph of a Max-CSP instance is the graph $\langle V, E \rangle$ where V is the set of variables and $\{X_i, X_j\} \in E$ if there is a pair of assignments p, q with $var(p) = X_i$, $var(q) = X_j$ and such that $cost(p, q) = 1$. Clearly the constraint graph of any instance in which E occurs contains a triangle. Max-Cut is NP-hard even on triangle-free graphs [17]. Any such instance of Max-Cut coded as an instance I of binary Max-CSP does not contain E as a subproblem since the constraint graph of I is triangle-free. Hence $\mathcal{F}(E)$ is intractable. □

Lemma 8. *The only 3-variable subproblem P for which the set $\mathcal{F}(P)$ is tractable is the subproblem C shown in Fig. 3.*

Proof. Let P be a 3-variable subproblem. For $\mathcal{F}(P)$ to be tractable, P must not have as a subproblem any of Q_0, Q_2, A, B, D, E which have all been shown to define intractable classes (Lemmas 4, 5, 7 and Corollary 1). The only 3-variable subproblem which does not contain any of Q_0, Q_2, A, B, D, E is C. The result then follows from Lemma 1. □

5 Subproblems on More Than Three Variables

It turns out that the tractable classes we have already identified, defined by forbidden subproblems on two or three variables, are the only possible tractable classes defined by forbidding a single subproblem. To complete our dichotomy, we require one final lemma.

Lemma 9. *If F is the subproblem shown in Fig. 3, then $\mathcal{F}(F)$ is intractable.*

Proof. It is known that Max-Cut on C_4-free graphs is NP-hard [16]. To see this, let G be a graph and G' a version of G in which each edge is replaced by a path composed of three edges. Clearly, G' is C_4-free and the maximum cut of G' is of the same size as the maximum cut of G.

When a Max-Cut instance on a C_4-free graph is coded as a Max-CSP instance I, the subproblem F cannot occur since there can be no length-4 cycles of non-trivial constraints in I. Hence $\mathcal{F}(F)$ is intractable. □

By looking at all possible combinations of edges in a subproblem, it is possible to show that F is the only subproblem on four variables in which neither A, B, D nor E shown in Fig. 3 occur. Since $\mathcal{F}(A)$, $\mathcal{F}(B)$, $\mathcal{F}(D)$ and $\mathcal{F}(E)$ are intractable, then from Lemma 9 the classes of Binary Max-CSP instances defined by forbidding a single subproblem on four or more variables are all intractable and we can now state our dichotomy.

Theorem 2. *If P is a binary Max-CSP subproblem, then $\mathcal{F}(P)$ is tractable if and only if P occurs either in Q_1 (shown in Fig. 2) or in C (shown in Fig. 3).*

It follows that $\mathcal{F}(P)$ is tractable only for $P = P_0$, P_1, Q_1 (shown in Fig. 2) or C (shown in Fig. 3). It follows that the only non-trivial tractable class defined by a forbidden subproblem corresponds to the set of instances satisfying the so-called joint-winner property. The joint-winner property encompasses, among other things, codings of non-intersecting graph-based or variable-based SoftAllDiff constraints together with arbitrary unary constraints [7]. It is worth pointing out that Theorem 2 is independent of the presence of unary cost functions, in the sense that tractable classes remain tractable when arbitrary unary costs are allowed and NP-hardness results are valid even if no unary costs are allowed.

6 Forbidding Sets of Subproblems

Certain known tractable classes can be defined by forbidding more than one subproblem. For example, in [5] it was shown that $\mathcal{F}(\{A, B\})$, $\mathcal{F}(\{B, D\})$ and $\mathcal{F}(\{A, D\})$ are all tractable (where A, B, C, D are the subproblems given in Fig. 3). The most interesting of these three tractable classes is $\mathcal{F}(\{A, B\})$ which is equivalent to maximum matching in graphs.

In this section we give a necessary condition for a forbidden set of subproblems to define a tractable class of binary Max-CSP instances.

Definition 1. *A subproblem (or an instance) P is* Boolean *if the size of the domain of each variable in P is at most two.*

A negative edge pair *is a set of variable-value assignments p, q, r, s such that $var(p) = var(r) \neq var(q) = var(s)$, $cost(p, q) = cost(r, s) = 1$ and $p \neq r$. A* positive edge pair *is a set of variable-value assignments p, q, r, s such that $var(p) = var(r) \neq var(q) = var(s)$, $cost(p, q) = cost(r, s) = 0$ and $p \neq r$.*

A negative cycle *is a set of variable-value assignments $p_1, \dots, p_m$, with $m > 2$, such that the variables $var(p_i)$ $(i = 1, \dots, m)$ are all distinct, $cost(p_i, p_{i+1}) = 1$ $(i = 1, \dots, m)$ and $cost(p_m, p_1) = 1$. A* positive cycle *is a set of assignments $p_1, \dots, p_m$ $(m > 2)$, such that the variables $var(p_i)$ $(i = 1, \dots, m)$ are all distinct, $cost(p_i, p_{i+1}) = 0$ $(i = 1, \dots, m)$ and $cost(p_m, p_1) = 0$.*

A negative pivot point *is a variable-value assignment p such that there are two assignments q, r with $var(p)$, $var(q)$, $var(s)$ all distinct and $cost(p, q) = cost(p, r) = 1$. A* positive pivot point *is an assignment p such that there are two assignments q, r with $var(p)$, $var(q)$, $var(s)$ all distinct and $cost(p, q) = cost(p, r) = 0$.*

Proposition 1. *If Σ is a finite set of subproblems, then $\mathcal{F}(\Sigma)$ is tractable only if*

1. *There is a Boolean subproblem $P \in \Sigma$ such that P contains no negative edge pair, no negative cycle and at most one negative pivot point, and*
2. *There is a Boolean subproblem $Q \in \Sigma$ such that Q contains no positive edge pair, no positive cycle and at most one positive pivot point, and*

3. *There is a Boolean subproblem $B \in \Sigma$ such that B contains neither Q_0 nor Q_2 (as shown in Fig. 2).*

Proof. Suppose that condition (1) is not satisfied. We will show that $\mathcal{F}(\Sigma)$ is NP-hard. Let t be an odd integer strictly greater than the number of variables in any subproblem in Σ. As in Lemma 5 the proof is by a polynomial reduction from Max-Cut. This time each constraint $X_i \neq X_j$ is replaced by the gadget G_t where G_t is $\neg X_i \vee Y_1$, $\neg Y_k \vee Y_{k+1}$ $(k = 1, \dots, t-1)$, $\neg Y_t \vee \neg X_j$, and $X_i \vee \neg Z_1$, $Z_k \vee \neg Z_{k+1}$ $(k = 1, \dots, t-1)$, $Z_t \vee X_j$. The gadget G_t is equivalent to $X_i \neq X_j$ since when $X_i = X_j$ one of its constraints must be violated, but when $X_i \neq X_j$ all of its constraints can be satisfied. For each pair of variables X, X' in the resulting instance of Max-CSP such that there is no constraint between X and X', we place a binary constraint on X, X' of constant cost 0.

The resulting instance I has no domain of size greater than two, and contains no negative edge pair, no negative cycle of length at most t and no two negative pivot points at a distance at most t. Let $P \in \Sigma$. Since (1) is not satisfied, and by definition of t, either P has a domain of size more than two, or contains a negative edge pair, a negative cycle of length at most t or two negative pivot points at a distance at most t. It follows that P cannot occur in I. Thus, we have demonstrated a polynomial reduction from Max-Cut to $\mathcal{F}(\Sigma)$.

The proof for condition (2) is similar. This time each constraint $X_i \neq X_j$ is replaced by the gadget G'_t given by $\neg X_i \wedge Y_1$, $\neg Y_k \wedge Y_{k+1}$ $(k = 1, \dots, t-1)$, $\neg Y_t \wedge \neg X_j$, and $X_i \wedge \neg Z_1$, $Z_k \wedge \neg Z_{k+1}$ $(k = 1, \dots, t)$, $Z_t \wedge X_j$. The gadget G'_t is equivalent to the constraint $X_i \neq X_j$; when $X_i = X_j$ at most t of its constraints can be satisfied and when $X_i \neq X_j$ at most $t+1$ of its constraints can be satisfied. For each pair of variables X, X' in the resulting instance of Max-CSP such that there is no constraint between X and X', we place a binary constraint on X, X' of constant cost 1.

The resulting instance I has no domain of size greater than two, and contains no positive edge pair, no positive cycle of length at most t and no two positive pivot points at a distance at most t. Let $P \in \Sigma$. If condition (2) is not satisfied, no $P \in \Sigma$ can occur in I. Thus, this polynomial reduction is from Max-Cut to $\mathcal{F}(\Sigma)$.

If condition (3) is not satisfied, then, by Lemma 1, $\mathcal{F}(\Sigma)$ contains all Boolean instances in $\mathcal{F}(Q_0, Q_2)$. But $\mathcal{F}(Q_0, Q_2)$ is equivalent to the set of Boolean instances of Max-CSP in which for each pair of variables X_i, X_j there is a constraint between X_i and X_j with this constraint being either $X_i = X_j$ or $X_i \neq X_j$. This set of Max-CSP instances is equivalent to the 2-Cluster Editing problem whose decision version is known to be NP-complete [20][3]. Hence $\mathcal{F}(\Sigma)$ is NP-hard if condition (3) is not satisfied. □

7 Conclusion

We have given a dichotomy concerning the tractability of classes of binary Max-CSP instances defined by forbidding a single subproblem. We have also given a

[3] We are grateful to Peter Jeavons for pointing out this equivalence.

necessary condition for the tractability of classes defined by forbidding sets of subproblems.

Classes defined by forbidding (sets of) subproblems are closed under permutations of the set of variables and independent permutations of each variable domain. An interesting avenue of future research is to place structure, such as an ordering, on the set of variables or on domain elements within the forbidden subproblems with the aim to uncover novel tractable classes.

References

1. Bertelé, U., Brioshi, F., *Nonserial dynamic programming*, Academic Press (1972).
2. Cohen, D. A., A New Class of Binary CSPs for which Arc-Constistency Is a Decision Procedure, *CP 2003*, LNCS 2833, Springer (2003), 807–811.
3. Cooper, M. C. and Escamocher, G., A Dichotomy for 2-Constraint Forbidden CSP Patterns, *AAAI 2012*.
4. Cooper, M.C., Jeavons, P.G. and Salamon, A.Z.: Generalizing constraint satisfaction on trees: hybrid tractability and variable elimination. *Artificial Intelligence* 174(9–10) (2010) 570–584.
5. Cooper, M. C. and Živný, S., Tractable triangles, *CP 2011*, LNCS 6876, Springer (2011) 195–209.
6. Cooper, M. C. and Živný, S., Hierarchically nested convex VCSP, *CP 2011*, LNCS 6876, Springer (2011) 187–194.
7. Cooper, M. C. and Živný, S., Hybrid tractability of valued constraint problems, *Artificial Intelligence* 175 (9-10) (2011) 1555–1569.
8. Creignou, N., Khanna, S. & Sudan, M. *Complexity classification of Boolean constraint satisfaction problems*, volume 7 of *SIAM Monographs on Discrete Mathematics and Applications* (2001).
9. Dalmau, V, Kolaitis, P.G. and Vardi, M.Y., Constraint satisfaction, bounded treewidth, and finite-variable logics, *CP 2002*, LNCS 2470, Springer (2002) 310–326.
10. Dechter, R., *Constraint Processing*, Morgan Kaufmann (2003).
11. Deineko, V., Jonsson, P., Klasson, M. and Krokhin, A., The approximability of Max CSP with fixed-value constraints, *Journal of the ACM* 55 (4) (2008).
12. Grohe, M., The complexity of homomorphism and constraint satisfaction problems seen from the other side, *Journal of the ACM* 54 (1) (2007) 1–24.
13. Jégou, P., Decomposition of Domains Based on the Micro-Structure of Finite Constraint-Satisfaction Problems, *AAAI '93*, Washington, D.C., (1993) 731–736.
14. Jonsson, P., Klasson, M. and Krokhin, A., The approximability of three-valued MAX CSP, *SIAM J. Comput.* 35 (6) (2006) 1329–1349.
15. Jonsson, P., Kuivinen, F. and Thapper, J., Min CSP on Four Elements: Moving Beyond Submodularity, *CP 2011* LNCS 6876, Springer (2011) 438–453.
16. Kamiński, M., MAX-CUT and containment relations in graphs, *WG 2010 - 36th International Workshop on Graph Theoretic Concepts in Computer Science* (2010) 15–26.
17. Lewis, J.M. and Yannakakis, M., The node-deletion problem for hereditary properties is NP-complete, *Journal of Computer System Sciences* 20(2) (1980) 219–230.
18. Orlin, J.B., A faster strongly polynomial time algorithm for submodular function minimization, *Mathematical Programming Ser. A* 118 (2) (2009) 237–251.

19. Salamon, A.Z. and Jeavons, P.G., Perfect Constraints Are Tractable, *CP 2008* LNCS 5202, Springer (2008) 524–528.
20. Shamir, R., Sharan, R. and Tsur, D., Cluster graph modification problems, *Discrete Applied Mathematics* 144 (2004) 173–182.