

available at www.sciencedirect.comjournal homepage: www.elsevier.com/locate/cosrev

Survey

Colouring, constraint satisfaction, and complexity

Pavol Hell^a, Jaroslav Nešetřil^{b,*}

^a School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6

^b KAM and ITI, Charles University, Malostranské náměstí 22, Prague, Czech Republic

ARTICLE INFO

Article history:

Received 2 October 2008

Received in revised form

7 October 2008

Accepted 7 October 2008

For Aleš Pultr on the Occasion of his
70th Birthday

ABSTRACT

Constraint satisfaction problems have enjoyed much attention since the early seventies, and in the last decade have become also a focus of attention amongst theoreticians. Graph colourings are a special class of constraint satisfaction problems; they offer a microcosm of many of the considerations that occur in constraint satisfaction. From the point of view of theory, they are well known to exhibit a dichotomy of complexity — the k -colouring problem is polynomial-time solvable when $k \leq 2$, and NP-complete when $k \geq 3$. Similar dichotomy has been proved for the class of graph homomorphism problems, which are intermediate problems between graph colouring and constraint satisfaction. However, for general constraint satisfaction problems, dichotomy has only been conjectured. Although the conjecture remains unproven to this day, it has been driving much of the theoretical research on constraint satisfaction problems, which combines methods of logic, universal algebra, analysis, and combinatorics. Currently, this is a very active area of research, and it is our goal here to present some of the recent developments, updating some of the information in existing books and surveys, while focusing on both the mathematical and the computational aspects of the theory. Given the level of activity, we are only able to survey a fraction of the new work, with emphasis on our own areas of interest.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

Consider the following scheduling application. Each semester at a typical university, the courses taught need to be scheduled for examination: several courses can be examined in one examination period, but two courses that have common students must be scheduled at different times. The university aims to schedule all examinations in as few examination periods as possible.

The situation as simplified above leads to a classical formulation via graph colouring [31,135,175]. We form the graph G in which the vertices are the courses taught, and

in which two courses are adjacent just if the courses *conflict*, i.e., have students in common. A schedule with k exam periods corresponds exactly to a k -colouring of G . In another view of this model, we may say that the courses represent *variables* with *values* from a certain *domain* of possibilities, which are *constrained* by requiring certain pairs of courses (those that conflict) to be assigned different values.

1.1. Constraint satisfaction

A general *constraint satisfaction problem* is given by a set \mathcal{V} of *variables* which are to be assigned *values* from a domain

* Corresponding author.

E-mail addresses: pavol@cs.sfu.ca (P. Hell), nesetřil@kam.mff.cuni.cz (J. Nešetřil).

\mathcal{D} , and a set \mathcal{C} of constraints, each of which restricts certain combinations of variables to certain sets of allowed values. Formally, a constraint satisfaction problem is defined as a triple $\mathcal{V}, \mathcal{D}, \mathcal{C}$, where \mathcal{V} is a set of variables, \mathcal{D} is a domain (set) of values, and \mathcal{C} is a set of constraints. Each constraint is a triple (r, t, U) , where r is a positive integer called *arity* of the constraint, t is an r -tuple of variables, and U is a set of r -tuples of values. An *evaluation* of the variables is a mapping $f : \mathcal{V} \rightarrow \mathcal{D}$. Such an evaluation satisfies a constraint $(r, t, U) \in \mathcal{C}$ if $(f(t_1), f(t_2), \dots, f(t_r)) \in U$. A *solution* is an evaluation that satisfies all constraints.

Obviously, such a general context allows much more refined models of the exam scheduling situation — we may express a much wider variety of restrictions on the schedule. For instance, suppose we wish to express the natural space limitation requiring that no more than three experimental physics exams are to be scheduled at any given time. This can be seen as a restriction on each quadruple of experimental physics courses, which limits their assigned values to those quadruples of time periods which do not fall within the same time period.

As another example, consider the *channel assignment problem* in which frequencies (channels) are assigned to transmitters for wireless communication, cf., e.g., [72]. Since the number of frequencies is limited, the operator must find a way to reuse the channels, in a way that minimizes interference. Interference will surely happen on two transmitters on the same antenna; these must receive different channels. Interference may also happen, to a smaller degree, on two transmitters on different antennas but in the same base transceiver station, or located in a particular way with respect to a geographical feature. Furthermore, there may be some interference even if two transmitters operate on adjacent, or nearby, channels, or on a channel and its harmonic. There may also be restrictions on which channels a particular transmitter may use — for instance at the edge of the operator's territory there may be constraints imposed by a neighbouring country or operator. We can model this problem by viewing the individual transmitters as variables whose values are the frequencies. To express the interference constraints we simply restrict pairs of more or less interfering transmitters to frequencies that can accommodate them. In addition to these binary constraints we also have the unary constraints that restrict the transmitters at the edge of the territory to frequencies available in such a location.

These examples should make it clear that a great variety of natural problems can be expressed in this model, including problems in scheduling, planning, data bases, machine vision, belief maintenance, temporal reasoning, type reconstruction, and many other areas of artificial intelligence [36,43,123,134,142,143,179,181]. The model was pioneered by Montanari [146], and enjoys wide popularity: there are journals entirely devoted to the solution of such problems, and books have been written about them [178]. The recent Handbook of Constraint Satisfaction [1] witnesses the activity in the field. Our focus here is on the theoretical aspects of constraint satisfaction, which forms only a small but nevertheless very active part of this area. (In the Handbook [1], it is represented by one chapter.) Due to this rapid development, we feel another update of recent surveys such as [69,96,103,97,147,149] is justified.

1.2. Homomorphisms

An alternate view of constraint satisfaction was proposed by Feder and Vardi [71]. Basically, it returns to the original definition of the graph colouring problem described earlier. In that case we were able to describe all constraints as a non-equality relation imposed on certain pairs of vertices. With a suitably generalized context, we will be able to do something similar for all constraint satisfaction problems.

A *relational structure* G consists of a finite set $V(G)$, whose elements we shall call *vertices* in order to underscore our graph theoretic inspiration, and a finite number of relations R_1, R_2, \dots, R_p , of arities r_1, r_2, \dots, r_p respectively. The vector (r_1, r_2, \dots, r_p) is called *the type* of G . Given two relational structures G (with vertex set $V(G)$ and relations R_1, R_2, \dots, R_p) and H (with vertex set $V(H)$ and relations S_1, S_2, \dots, S_p), of the same type (the arity of each R_i is the same as that of the corresponding S_i), a *homomorphism* of G to H is a mapping $f : V(G) \rightarrow V(H)$ which preserves all pairs of corresponding relations, i.e., such that $(v_1, v_2, \dots, v_{r_i}) \in R_i$ implies $(f(v_1), f(v_2), \dots, f(v_{r_i})) \in S_i$, for all $i = 1, 2, \dots, p$.

In order to minimize the notation, we have adopted a natural extension of the graph theoretic notation. However, relational structures are complex objects, and we remark that it is more common to use a special notation, where a structure is written in bold font, say \mathbf{A} , its ground set by the corresponding letter, A , and the relational symbols are usually written as R_A .

Given a constraint satisfaction problem with variables \mathcal{V} , domain \mathcal{D} , and a set \mathcal{C} of constraints, we define the structure G with $V(G) = \mathcal{V}$ and the structure H with $V(H) = \mathcal{D}$, where each U occurring in some constraint $C = (r, t, U)$ defines an r -ary relation S_i on $V(H)$, for which the corresponding relation R_i consists of all r -tuples t' of variables with a constraint $C' = (r, t', U)$. Then a solution to the constraint satisfaction problem is simply a homomorphism of G to H . Of course, the reverse of this transformation succeeds in expressing the existence of any homomorphism of some general structures G to H as a constraint satisfaction problem. Therefore from now on we think of constraint satisfaction problems as problems seeking the existence of a homomorphism between relational structures [71]. This is not to say that we advocate the use of such reformulations in individual constraint satisfaction problems, quite to the contrary: the relational formulation does not have the same “feel” for individual data and its semantic meaning which can (and must) be used in solving concrete problems. On the theoretical side, however, this formulation of constraint satisfaction problems has proved crucial, by relating it to the techniques of categorial and universal algebra [27,28] and more generally to algebraic combinatorics, as well as to statistical physics [2,19,76,102,103,154]. For example, the study of homomorphisms between relational structures has a long tradition in the ‘Prague School’ of category theory [103], and many theoretical insights, such as those recorded for instance in the monograph [169], are applicable to the present context.

A typical aspect of our example scheduling problem is the fact that the structure G of courses to be examined changes every semester, while the structure H of exam periods tends

to be much more stable. (For instance, which courses are taught, and which pairs of courses have students in common, vary, while the total number of exam periods, the number of periods per day, and so on, may remain constant for several consecutive semesters.) A similar situation arises in the channel assignment problem — again the values (frequencies) H are the same over a number of problems, while the variables (transmitters) G vary according to concrete locations. This occurs for many other applications (the structure H tends to be fixed), and it motivates the following variant of the constraint satisfaction problem.

Let H be a fixed relational structure. The *constraint satisfaction problem* $\text{CSP}(H)$ asks whether or not an input relational structure G , of the same type as H , admits a homomorphism to H .

Consider the problem $\text{CSP}(H)$, where H has only one (binary) relation, S_1 , consisting of all ordered pairs of distinct vertices of $V(H)$. Since the fixed structure H has one symmetric binary relation, the input structure G may be assumed to have also just one symmetric binary relation, i.e., G is a graph. Thus we have the problem of colouring the input structure (graph) G with $|V(H)|$ colours, so that adjacent vertices of G obtain different colours. This is the *graph colouring problem*. Suppose more generally that H has any one symmetric binary relation, in other words, H is an undirected graph. Then the problem $\text{CSP}(H)$ may be assumed to be restricted to undirected input graphs G ; it asks whether or not G admits a homomorphism to H . This problem is called the *graph H -colouring problem*. Similarly, if H is a structure with one (not necessarily symmetric) binary relation, i.e., a digraph, each input G is also a digraph, and the problem reduces to the existence of digraph homomorphisms [103]. We emphasize that this view regards undirected graphs as a special class of digraphs, namely ones with a symmetric adjacency relation. Many (but not all) phenomena that occur for general constraint satisfaction are typified by their restrictions in the context of digraph homomorphisms. Definitions and results are often stated in the restricted domain of digraph homomorphisms [103], but with the understanding that they apply equally well in the general context of constraint satisfaction. Consider, for instance, the following observations [103]. When H and H' are two digraphs (more generally, relational structures of the same type) which admit a homomorphism of H to H' as well as a homomorphism of H' to H , then the two problems $\text{CSP}(H)$ and $\text{CSP}(H')$ are obviously equivalent. (A structure G admits a homomorphism to H if and only if it admits a homomorphism to H' , since the composition of two homomorphisms is again a homomorphism.) A digraph (relational structure) H with no proper substructure H' to which H admits a homomorphism is called a *core* [99,103]. It is easy to check that every digraph (relational structure) H has, up to isomorphism, a unique core subdigraph (substructure) H' to which it admits a homomorphism [99,103]. Thus we typically restrict our attention to problems $\text{CSP}(H)$ where H is a core.

2. Dichotomy

The k -colouring problem is polynomial-time solvable when $k = 1, 2$ and is NP-complete otherwise. This basic fact

(established at the very onset of the theory of NP-completeness [79]) illustrates the *dichotomy* of possible complexities of the class of k -colouring problems, as k varies. There is, in principle, no reason for each colouring problem to be polynomial-time solvable (one of the easiest problems in NP) or NP-complete (one of the hardest problems in NP). Indeed, Ladner [127] has shown that if $P \neq \text{NP}$, there are problems in NP that are neither polynomial nor NP-complete — in fact there must be an infinite hierarchy of such (non-polynomially-equivalent) problems. Since these “intermediate difficulty” problems must exist in NP (unless $P = \text{NP}$), dichotomies are always somewhat surprising, especially if they apply to a broad class of problems.

Dichotomy for graph H -colouring problems has been proved by the authors in [101] (solving a problem from [117]). In other words, we have shown that, for every graph H , the problem of deciding the existence of a homomorphism of an input graph G to H is polynomial-time solvable, or is NP-complete. In fact, we have classified which problems are polynomial-time solvable and which are NP-complete.

Theorem 2.1 (*Graph Dichotomy* [101]). *Suppose H is a graph, i.e., a relational structure with a single relation which is binary and symmetric. Then $\text{CSP}(H)$ is NP-complete, except in the following, polynomial-time solvable, cases.*

1. H is bipartite; or
2. H has a loop.

The proof of the Graph Dichotomy in [101] was surprisingly complex. In the intervening years, a number of new proofs have appeared [25,125,183], based on a great variety of ideas and approaches. We discuss these in several places of this manuscript, as they present a testing ground for these techniques. In this sense, the Graph Dichotomy is a leitmotif of this survey.

Another early dichotomy has been proved for all Boolean-satisfiability problems by Schaeffer [176]. These are the problems $\text{CSP}(H)$ where H has two vertices, say, 0 and 1. To describe Schaeffer’s classification, we shall recall four well-known operations on tuples. The OR operation on two tuples (a_1, a_2, \dots, a_s) and (b_1, b_2, \dots, b_s) is the tuple (z_1, z_2, \dots, z_s) where each $z_i = a_i \vee b_i$ ($z_i = 1$ unless both $a_i = b_i = 0$, in which case $z_i = 0$). The AND operation on two tuples (a_1, a_2, \dots, a_s) and (b_1, b_2, \dots, b_s) is the tuple (z_1, z_2, \dots, z_s) where each $z_i = a_i \wedge b_i$ ($z_i = 0$ unless both $a_i = b_i = 1$, in which case $z_i = 1$). The MAJORITY operation on three tuples (a_1, a_2, \dots, a_s) , (b_1, b_2, \dots, b_s) , and (c_1, c_2, \dots, c_s) is the tuple (z_1, z_2, \dots, z_s) where each z_i is the majority value (0 or 1) of a_i, b_i, c_i . The XOR (exclusive OR, also known as MINORITY) operation on three tuples (a_1, a_2, \dots, a_s) , (b_1, b_2, \dots, b_s) , and (c_1, c_2, \dots, c_s) is the tuple (z_1, z_2, \dots, z_s) where each z_i is the exclusive-OR value of a_i, b_i, c_i (equal to 1 if the number of 1’s amongst a_i, b_i, c_i is odd, and 0 otherwise). Schaeffer proved the following classification [176].

Theorem 2.2 (*Boolean Dichotomy* [176]). *Suppose H is a relational structure with $V(H) = \{0, 1\}$ and relations S_1, S_2, \dots, S_p . Then $\text{CSP}(H)$ is NP-complete, except in the following, polynomial-time solvable, cases:*

1. each S_i contains the s_i -tuple $(0, 0, \dots, 0)$; or
2. each S_i contains the s_i -tuple $(1, 1, \dots, 1)$; or

3. each S_i is closed under the OR operation; or
4. each S_i is closed under the AND operation; or
5. each S_i is closed under the MAJORITY operation; or
6. each S_i is closed under the XOR operation.

The polynomial-time algorithms for these cases are well known. In cases 1 and 2, the core of H has one vertex, and any structure G admits a homomorphism to this core (and hence to H). Case 3 (respectively 4) corresponds to problems equivalent to the case where each S_i consists of all s_i -tuples with 1 in the first coordinate, plus possibly the tuple $(0, 0, \dots, 0)$ (respectively all s_i -tuples with 0 in the first coordinate, plus possibly the tuple $(1, 1, \dots, 1)$). Thus they can be expressed by *Horn clauses* (respectively *dual-Horn clauses*), i.e., disjunctions with at most one negated (respectively unnegated) variable, and solved as in [46,108]. Case 5 corresponds to problems equivalent to H having just four binary relations, S_1 consisting of all pairs other than $(0, 0)$, S_2 consisting of all pairs other than $(0, 1)$, S_3 , consisting of all pairs other than $(1, 0)$, and S_4 , consisting of all pairs other than $(1, 1)$. Thus these are the problems expressible by disjunctions with two variables each, i.e., by 2-satisfiability [6]. The last case corresponds to systems of linear equations modulo two, solvable by Gaussian elimination.

The Boolean and Graph Dichotomy theorems, Theorems 2.1 and 2.2 motivated Feder and Vardi [71] to formulate the following conjecture, which remains open to this day, and motivates much research in the area.

Conjecture 2.3 (The Dichotomy Conjecture [71]). *For any relational structure H , the problem $\text{CSP}(H)$ is NP-complete or polynomial-time solvable.*

By now there is strong supporting evidence for the conjecture — in the intervening years it has been verified in many cases [11–13,22,23,14,37,39,49,51–53,58–61,65,70,133,137,168], cf. [50,103], prominently including $\text{CSP}(H)$ for structures H with up to three vertices [24], extending the Boolean Dichotomy of [176], and for conservative structures [22], discussed in Section 5.

It is important to note that dichotomy is *not* known for the case of *digraphs*, i.e., for problems $\text{CSP}(H)$ where H has only one relation, S_1 , which is binary (but not necessarily symmetric). Many results have been proved, classifying the complexity of these *digraph H -colouring problems* for special families of digraphs H [11,14,12,13,49,65,90]. In fact, [11,14] conjectured in 1989, a specific dichotomy classification for digraphs without sources and sinks (i.e., with all indegrees and outdegrees positive).

Conjecture 2.4 ([11,14]). *Suppose H is a core digraph with all indegrees and outdegrees positive. If each component of H is a directed cycle, the digraph H -colouring problem is polynomial-time solvable; otherwise it is NP-complete.*

After nearly two decades, this conjecture has been proved [15]. (See Section 3.3.) The fact that this longstanding open problem could now be solved, testifies to the success of the algebraic method outlined in the following sections.

With vertices of indegree or outdegree zero, very little is known. For instance, dichotomy is not known if the underlying undirected graph of H is a tree, not even if it is a

union of three paths meeting at one vertex — for the so-called *triads* H [104].

Finally, we note that it was shown by Feder and Vardi [71] that dichotomy for digraph H -colouring problems would imply the entire Dichotomy Conjecture 2.3. Thus there is a striking difference between the H -colouring problems for graphs and for digraphs.

2.1. A combinatorial view of NP

The Dichotomy Conjecture 2.3 appears to be an important and difficult question. In a sense, the class of problems $\text{CSP}(H)$ is a largest class in which dichotomy can be expected. This was concretely formulated and proved by Feder and Vardi [71]. Recently, the formulation has been combinatorially refined by Kun and Nešetřil [126]; we now review this research. The logic class SNP (“syntactic NP”) consists of all problems expressible by an existential second-order formula with a universal first-order part [48,71,120]. For our purposes, we view the input of the problem as a relational structure S with the vertex set $V(S) = X$ and relational symbols $R(S) = R$, and view the existentially quantified relations as “proof relations” Π . It is shown in [71] that every problem (language) L in SNP is equivalent to a formula of the form

$$\exists P \forall \bar{x} \in S \bigwedge_i \neg (\alpha_i \wedge \beta_i \wedge \varepsilon_i),$$

where

- α_i is a conjunction of atoms or negated atoms involving variables and input relations (i.e., is of the form $R(\bar{x})$ and $\neg R(\bar{x})$ for a relational symbol R and \bar{x} a tuple of elements of X);
- β_i is a conjunction of atoms and negated atoms involving variables and existentially quantified proof relations (i.e., is of the form $P(\bar{x})$ and $\neg P(\bar{x})$ for $P \in \Pi$ and \bar{x} a tuple of elements of X); and
- ε_i is the conjunction of atoms involving variables and inequalities (i.e. of the form $x \neq y$).

A formula of this type is called a *canonical formula* of the language L . We say that the language L is *monotone* if there are no negations in the α_i 's. (In such a language, more relations lead to fewer satisfiable formulas.) The language L is *monadic* if the relations in P are all monadic (all proof relations are unary). The language L is *without inequality* if no ε_i appears in the formula.

Example. Consider the language containing one binary symbol R and two unary proof relations P_1, P_2 and the following formula

$$\begin{aligned} \exists P_1 \exists P_2 \forall x_1, x_2, x_3, y \in X \\ & (P_1(x_1) \wedge P_1(x_2) \wedge P_1(x_3)) \vee (P_2(x_1) \wedge P_2(x_2) \\ & \wedge P_2(x_3)) \wedge (R(x_1, x_2) \wedge R(x_1, x_3) \wedge R(x_2, x_3)) \\ & \wedge ((x_1 \neq x_2) \wedge (x_1 \neq x_3) \wedge (x_2 \neq x_3)) \wedge [\neg(\neg P_1(y) \wedge \neg P_2(y))]. \end{aligned}$$

This formula corresponds to the language of all relations whose vertices can be covered by two sets in such a way that neither of these sets contains a triple linearly ordered by R . If we in addition postulate that the relation R is symmetric then these are just graphs which can be vertex partitioned into two triangle-free graphs.

