

Locally Consistent Constraint Satisfaction Problems with Binary Constraints

Manuel Bodirsky¹ and Daniel Král'^{2,3*}

¹ Institut für Informatik, Abteilung Algorithmen und Komplexität I,
Humboldt-Universität zu Berlin,
Unter den Linden 6, 10099 Berlin, Germany.
`bodirsky@informatik.hu-berlin.de`

² Institute for Mathematics, Technical University Berlin [†],
Strasse des 17. Juni 136, 10623 Berlin, Germany.

³ Department of Applied Mathematics,
Faculty of Mathematics and Physics, Charles University,
Malostranské náměstí 25, 118 00 Praha 1, Czech Republic.
`kral@kam.mff.cuni.cz`

Abstract. An instance of a constraint satisfaction problem is called *k-consistent* if any *k* constraints of it can be simultaneously satisfied. We focus on constraint languages with a single binary constraint. In this case, the constraint satisfaction problem is equivalent to the question whether there is a homomorphism from an input digraph *G* to a fixed target digraph *H*. The instance corresponding to *G* is *k-consistent* if every subgraph of *G* of size at most *k* is homomorphic to *H*. Let $\rho_k(H)$ be the largest ρ such that every *k-consistent* *G* contains a subgraph *G'* of size at least $\rho||E(G)||$ that is homomorphic to *H*. The ratio $\rho_k(H)$ reflects the fraction of constraints of a *k-consistent* instance that can be always satisfied. We determine $\rho_k(H)$ for all digraphs *H* that are not acyclic and show that $\lim_{k \rightarrow \infty} \rho_k(H) = 1$ if *H* has tree duality. For the latter case we design an algorithm that computes in linear time for a given input graph *G* either a homomorphism from almost the entire graph *G* to *H*, or a subgraph of *G* of bounded size that is not homomorphic to *H*.

1 Introduction

Constraint satisfaction problems (CSP) form an important model for problems arising in many areas of computer science. This is witnessed by the interest in the computational complexity of various variants of CSP's [1, 2, 8–11, 23]. However, sometimes not all the constraints need to be satisfied, but it suffices to satisfy a large fraction of them. In order to maximize this fraction, the input can be pruned at the beginning by removing small sets of contradictory constraints so

* Supported by Institute for Theoretical Computer Science (ITI), project 1M0021620808 of Czech Ministry of Education.

[†] The author is a postdoctoral fellow at TU Berlin within the framework of the European training network COMBSTRU.

that the input becomes “locally” consistent. Formally, an instance of CSP is said to be *k-consistent* if any k constraints can be simultaneously satisfied.

A similar notion of local consistency can be defined in terms of variables: an instance is k -consistent if the values of any k variables can be chosen so that any constraint on only these variables is satisfied. Our results extend to this setting.

Both these notions of local consistency differ fundamentally from the notion of *k-consistency* of Freuder [11] (and also the notion of *relational k-consistency* of Dechter and van Beek [4]), where a CSP instance is *k-consistent* if every solution for the constraints on $k - 1$ variables (constraints) can be extended to another variable (constraint).

1.1 History of Locally Consistent CSPs

The notion of local consistency considered in this paper can be traced back to the early 1980’s. Lieberherr and Specker [19, 20] studied the problem for CNF formulas: they require that any k clauses of a given formula can be satisfied and asked what fraction of all the clauses can be satisfied. They settled the case $k = 1, 2, 3$. A simpler proof of their results was found by Yannakakis [24]. The case $k = 4$ was settled in [17] (exploring a connection to Usiskin’s numbers [21]). Locally consistent CNF formulas can also be found in Chapter 20 of [16].

Huang et al. [14] and Trevisan [22] resolved the asymptotic behavior of locally consistent CNF formulas as k approaches infinity. Trevisan [22] was the first to define the notion of local consistency for CSPs with constraints that are Boolean predicates. For a set Π of Boolean constraints, $\rho_k(\Pi)$ is the maximum ρ such that a fraction of at least ρ constraints can be satisfied in any k -consistent input. Note that we now allow negations in the arguments of the constraints (the domain is the Boolean field). If Π is the set of all the predicates of arity ℓ , then $\lim_{k \rightarrow \infty} \rho_k(\Pi) = 2^{1-\ell}$ [22]. The ratios $\rho_k(\Pi)$, $k \geq 1$, for a set Π consisting of a single predicate of arity at most three were determined by Dvořák et al. [6]. The asymptotic behavior for sets Π of predicates was studied in [18], where $\lim_{k \rightarrow \infty} \rho_k(\Pi)$ was expressed as the minimum of a certain functional on a convex set of polynomials derived from Π . Efficient algorithms for locally consistent CSPs with constraints that are Boolean predicates were also designed [6, 7, 18].

1.2 Our Contribution

We initiate the study of locally consistent CSPs on larger finite domains and focus on the case where all the constraints are of the same binary relation. The relation can be described by a digraph H whose vertices correspond to the elements of the domain. Two vertices are joined by an arc if the ordered pair of the corresponding elements is contained in the relation. Similarly, the input can be described by a digraph G : the vertices of G correspond to the variables and the arcs to the given constraints. There is a satisfying assignment for the input if and only if G is *homomorphic* to H , i.e., there exists a mapping $h : V(G) \rightarrow V(H)$ such that $h(u)h(v) \in E(H)$ for every $uv \in E(G)$.

The notion of local consistency translates to digraphs as follows: G corresponds to an k -consistent input if every subgraph of G of size at most k , i.e., with at most k edges, is homomorphic to H . The ratio $\rho_k(H)$ denotes the largest ρ such that for any k -consistent G there is a mapping $h : V(G) \rightarrow V(H)$ preserving at least $\rho|G|$ arcs of G . The version defined in terms of variables also translates to digraphs: for that we require that each subgraph G' of order at most k is homomorphic to H . The corresponding ratio is denoted by $\rho_k^v(H)$.

We can restrict our attention to digraphs H that are cores. H is a *core* if it does not have a homomorphism to a proper subgraph. Every digraph H contains a unique (up to isomorphism) subgraph H' such that H is homomorphic to H' and H' is a core. Obviously, $\rho_k(H) = \rho_k(H')$ and $\rho_k^v(H) = \rho_k^v(H')$.

We show that if H contains a directed cycle (or a loop), then $\rho_k(H)$ and $\rho_k^v(H)$ coincide and are equal to the fractional relative density $\delta'_{\text{rel}}(H)$ of H as defined in Section 2. For such H we design a simple linear time algorithm that finds a mapping $h : V(G) \rightarrow V(H)$ preserving at least $\delta'_{\text{rel}}(H) \cdot |G|$ arcs of G .

Finally, we focus on acyclic digraphs H . Using the notion of tree duality from [13] we show that $\lim_{k \rightarrow \infty} \rho_k(H) = 1$ for all orientations of a path and all acyclic tournaments. In general, the equality $\lim_{k \rightarrow \infty} \rho_k(H) = 1$ holds for all H that have tree duality. Unfortunately, there are acyclic digraphs H that do not have tree duality (there are even examples that are orientations of trees)—we discuss the cases not settled in a satisfactory way in the concluding Section 5 where we also mention possible generalizations for CSPs with larger languages.

2 Target Graphs with Cycles or Loops

We first focus on the binary relations (constraints) whose corresponding target graph H is not acyclic. This includes the case when the relation is symmetric.

For a digraph H , we define the *fractional relative density* of H as follows.

$$\delta'_{\text{rel}}(H) = \max_{x:V(H) \rightarrow \langle 0,1 \rangle} \sum_{v \in V(H)} x(v) \sum_{uv \in E(H)} x(u) \cdot x(v)$$

The maximum is taken over all functions $x : V(H) \rightarrow \langle 0, 1 \rangle$ such that the sum of $x(v)$ is equal to one. In particular, if H contains a loop, then $\delta'_{\text{rel}}(H) = 1$. This notion of density is similar to that of *relative density* as used e.g. in [15]:

$$\delta_{\text{rel}}(H) = \max_{\emptyset \neq H' \subseteq H} \frac{|H'|}{|H'|^2}.$$

The two notions are different in general. Consider the digraph H obtained by replacing each edge of K_5 by a bigon and removing two non-incident arcs. The relative density $\delta_{\text{rel}}(H)$ is $18/25 = 0.720$ but the fractional relative density $\delta'_{\text{rel}}(H)$ is $88/121 \approx 0.727$ (set $x(v) = 3/11$ for the vertex incident with 8 arcs and $x(v) = 2/11$ for the other vertices). However, the two notions coincide when H corresponds to a symmetric binary relation [5]. In this case, both the target and

the input graph can be viewed as an undirected digraph (replace each bigon by a single undirected edge).

Let $\omega(H)$ denote the size of the largest subset $A \subseteq V(H)$ such that any two distinct vertices of A are joined by an arc.

Proposition 1. *If a digraph H corresponds to a symmetric binary relation, then $\delta'_{\text{rel}}(H) = \delta_{\text{rel}}(H)$. Moreover, $\delta_{\text{rel}}(H)$ equals 1, if H contains a loop, and $\delta_{\text{rel}}(H) = 1 - 1/\omega(H)$ otherwise.*

Proof. If H contains a loop, then $\delta_{\text{rel}}(H) = \delta'_{\text{rel}}(H) = 1$. In the rest, we assume that H has no loops. Let $x : V(H) \rightarrow \langle 0, 1 \rangle$ be the function such that $\delta'_{\text{rel}}(H) = \sum_{uv \in E(H)} x(u) \cdot x(v)$, $\sum_{v \in V(H)} x(v) = 1$ and the support of x is minimal. We show that $uv \in E(H)$ for any two vertices u and v contained in the support.

Assume the opposite and let u and v be two non-adjacent vertices such that $x(u) > 0$ and $x(v) > 0$. Let $X_u = 2 \sum_{uw \in E(H)} x(w)$ and $X_v = \sum_{uw \in E(H)} x(w)$. By symmetry, we can assume that $X_u \leq X_v$. Consider the following function x' :

$$x'(w) = \begin{cases} x(u) - \min\{x(u), x(v)\} & \text{if } w = u, \\ x(v) + \min\{x(u), x(v)\} & \text{if } w = v, \text{ and} \\ x(w) & \text{otherwise.} \end{cases}$$

Since the vertices u and v are non-adjacent in H , the following holds:

$$\sum_{uv \in E(H)} x'(u) \cdot x'(v) - \sum_{uv \in E(H)} x(u) \cdot x(v) = 2(X_v - X_u) \min\{x(u), x(v)\}.$$

The choice of x implies that $X_u = X_v$ and $\delta'_{\text{rel}}(H) = \sum_{uv \in E(H)} x'(u) \cdot x'(v)$. Since $X_u = X_v$, the configuration is again symmetric with respect to u and v and we may assume that $x(u) < x(v)$. Consequently, $x'(u) = 0$ and the support of x is not minimal. We conclude that the support of x induces a complete graph.

It is an easy exercise in calculus to show that if $\delta'_{\text{rel}}(H) = \sum_{uv \in E(H)} x(u) \cdot x(v)$, then $x(u) = 1/k$ where k is the size of the support of x . Hence, $\delta'_{\text{rel}}(H) = 1 - 1/\omega(H)$. Since $\delta_{\text{rel}}(H) \leq \delta'_{\text{rel}}(H)$, we have $\delta_{\text{rel}}(H) = \delta'_{\text{rel}}(H)$.

The density $\delta'_{\text{rel}}(H)$ is always a lower bound on $\rho_k(H)$ (even if H is acyclic):

Lemma 1. *Let H be a digraph. The following holds for every $k \geq 1$:*

$$\rho_k(H) \geq \delta'_{\text{rel}}(H) .$$

Moreover, there exists a deterministic algorithm that for any digraph G finds a mapping $h : V(G) \rightarrow V(H)$ that preserves at least $\delta'_{\text{rel}}(H) \cdot \|G\|$ arcs of G . The running time of the algorithm is linear in the size of G (if H is fixed).

Proof. Let $x : V(H) \rightarrow \langle 0, 1 \rangle$ be the function such that $\sum_{v \in V(H)} x(v) = 1$ and $\delta'_{\text{rel}}(H) = \sum_{uv \in E(H)} x(u) \cdot x(v)$. Consider a mapping $h : V(G) \rightarrow V(H)$ that maps each vertex of G to a vertex $v \in V(H)$ with probability $x(v)$. The probability that an arc of G is mapped to an arc of H is exactly $\sum_{uv \in E(H)} x(u) \cdot x(v) = \delta'_{\text{rel}}(H)$. Hence, the expected number of arcs preserved by h is $\delta'_{\text{rel}}(H) \cdot \|G\|$. The mapping h can be found deterministically in linear time using the derandomization method based on conditional expectations as described in [24].

We now recall Markov's inequality and Chernoff's inequality:

Proposition 2. *If X is a non-negative random variable with the expected value E , then the following holds for every $\alpha \geq 1$:*

$$\text{Prob}(X \geq \alpha) \leq \frac{E}{\alpha}.$$

Proposition 3. *If X is the sum of N independent random zero-one variables, each equal to one with probability p , then the following holds for $0 < \delta \leq 1$:*

$$\text{Prob}(X \geq (1 + \delta)pN) \leq e^{-\frac{\delta^2 pN}{3}} \quad \text{and} \quad \text{Prob}(X \leq (1 - \delta)pN) \leq e^{-\frac{\delta^2 pN}{2}}.$$

We now prove the converse inequality of Lemma 1:

Theorem 1. *If H is a non-acyclic digraph, then $\rho_k(H) = \delta'_{\text{rel}}(H)$ for all $k \geq 1$.*

Proof. Fix $k \geq 1$ and ε , $0 < \varepsilon \leq 1/2$. Let n be a sufficiently large integer. We find a digraph G whose every subgraph of size at most k is homomorphic to H , but every mapping $h : V(G) \rightarrow V(H)$ preserves at most $(\delta'_{\text{rel}}(H) + \varepsilon)|G|$ arcs.

We first consider a random digraph G_0 and we later prune it to obey all the constraints. G_0 is a random graph of order n in which the arc from u to v , $u \neq v$, is included with probability $n^{-1+1/2k}$ independently of the other arcs. G_0 contains no loops. Since the expected number of arcs of G_0 is $n(n-1)n^{-1+1/2k} = n^{1/2k}(n-1)$, Proposition 3 implies that the probability that the number of arcs is smaller than $(1 - \varepsilon/4)n^{1+1/2k}$ does not exceed $1/4$ for a sufficiently large n .

Next, we estimate the number of (not necessarily directed) cycles of G_0 . The expected number of bigons is $\binom{n}{2}n^{-2+2/2k} \leq n^{1/k}$ and that of cycles of length $\ell = 3, \dots, k$ is at most $n^\ell 2^\ell (n^{-1+1/2k})^\ell \leq 2^\ell n^{1/2}$. By Proposition 2, the number of such bigons and cycles does not exceed $4 \cdot k 2^k n^{1/2}$ with probability at least $1/4$. Hence, if n is sufficiently large (and k is fixed), the number of arcs contained in such bigons and cycles is bounded by $\varepsilon n/4$ with probability at least $3/4$.

Fix a mapping $h : V(G_0) \rightarrow V(H)$. Set $x(v) := |h^{-1}(v)|/n$ for $v \in V(H)$. By the definition of $\delta'_{\text{rel}}(H)$, the following holds:

$$\sum_{uv \in E(H)} x(u)x(v) \leq \delta'_{\text{rel}}(H) \tag{1}$$

The expected number of arcs of G_0 preserved by h can be estimated using (1):

$$\sum_{uv \in E(H)} |h^{-1}(u)| \cdot |h^{-1}(v)| n^{-1+1/2k} = \sum_{uv \in E(H)} x(u)x(v) n^{1+1/2k} \leq \delta'_{\text{rel}}(H) n^{1+1/2k}.$$

By Proposition 3, the probability that the number of arcs preserved by h exceeds $(1 + \varepsilon/4)\delta'_{\text{rel}}(H)n^{1+1/2k}$ is at most $e^{-\frac{\varepsilon^2 \delta'_{\text{rel}}(H) n^{1+1/2k}}{48}}$. Since there are $|V(H)|^n$ possible choices of h , and since the target graph H and the numbers k and ε are fixed, the probability that there exists $h : V(G_0) \rightarrow V(H)$ preserving more than $(1 + \varepsilon/4)\delta'_{\text{rel}}(H)n^{1+1/2k}$ arcs of G_0 is at most $1/4$ if n is sufficiently large.

We conclude that the following holds with a positive probability:

1. G_0 contains at least $(1 - \varepsilon/4)n^{1+1/2k}$ arcs,
2. the size of the set E of the arcs contained in bigons or cycles of length at most k in G_0 does not exceed $\varepsilon n/4$, and
3. every $h : V(G_0) \rightarrow V(H)$ preserves at most $(1 + \varepsilon/4)\delta'_{\text{rel}}(H)n^{1+1/2k}$ arcs.

Therefore, there exists a graph G_0 with the above three properties. The final graph G is obtained from G_0 by removing the arcs contained in the set E .

The size of G is at least $(1 - \varepsilon/2)n^{1+1/2k}$. Since every mapping $h : V(G_0) \rightarrow V(H)$ preserves at most $(1 + \varepsilon/4)\delta'_{\text{rel}}(H)n^{1+1/2k}$ arcs and G is a subgraph of G_0 , every $h : V(G) \rightarrow V(H)$ also preserves at most this number of arcs. We now infer the bound on the fraction of arcs preserved by h (recall that $\varepsilon \leq 1/2$):

$$\frac{(1 + \varepsilon/4)\delta'_{\text{rel}}(H)n^{1+1/2k}}{\|G\|} \leq \frac{1 + \varepsilon/4}{1 - \varepsilon/2}\delta'_{\text{rel}}(H) \leq (1 + \varepsilon)\delta'_{\text{rel}}(H).$$

Next, we show that any subgraph of G of size at most k is homomorphic to H . Let G' be such a subgraph. Since the size of G' is at most k , G' is an orientation of a forest. Hence, there is a homomorphism from G' to any directed cycle. In particular, there is a homomorphism from G' to H .

Since for every $\varepsilon > 0$, there is a digraph G corresponding to a k -consistent input and the fraction of arcs preserved by any $h : V(G) \rightarrow V(H)$ is at most $\delta'_{\text{rel}}(H) + \varepsilon$, we have $\rho_k(H) \leq \delta'_{\text{rel}}(H)$. The opposite inequality holds by Lemma 1.

Proposition 1 and Theorem 1 imply the following for symmetric relations:

Corollary 1. *Let H be a digraph corresponding to a symmetric binary relation \mathcal{R} . The following holds for every $k \geq 1$:*

$$\rho_k(H) = \begin{cases} 1 & \text{if there exists an element } a \text{ such that } [a, a] \in \mathcal{R} \\ 1 - 1/\omega(H) & \text{otherwise} \end{cases}$$

Since every subgraph of order at most k of G from the proof of Theorem 1 is an orientation of a forest and thus homomorphic to H , the following also holds:

Corollary 2. *If H is a non-acyclic digraph, then $\rho_k^v(H) = \delta'_{\text{rel}}(H)$ for all $k \geq 1$.*

3 Graph Homomorphisms and Tree Duality

A key ingredient to our algorithm in the next section is the notion of tree duality. A digraph H has *tree duality* if G is homomorphic to H if and only if every directed tree homomorphic to G is also homomorphic to H . E.g., every orientation of a simple path or every acyclic tournament has tree duality. Feder and Vardi [10] and Hell, Nešetřil and Zhu [13] observed that if H has tree duality, then the H -coloring problem (the decision problem whether a given graph is homomorphic to H) can be solved in polynomial time [12] by the so-called *arc-consistency procedure* (also called *consistency check*).

An equivalent definition of having tree duality uses the notion of set graphs. For a digraph H , the *set graph* 2^H is the graph whose vertices are non-empty

subsets of $V(H)$ and two subsets U and V are joined by an arc if the following holds: for every vertex $u \in U$, there exists a vertex $v \in V$ such that uv is an arc of H , and for every vertex $v \in V$, there exists a vertex $u \in U$ such that uv is an arc of H (see Figure 1). H has tree duality if and only if 2^H is homomorphic to H [3, 10]. Note that this criterion can be used to decide algorithmically whether a given digraph H has tree duality.

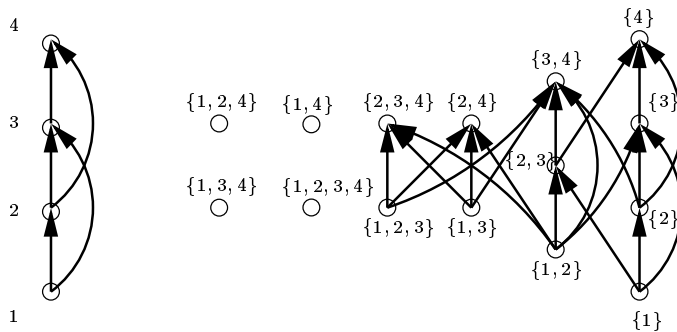


Fig. 1. An example of a digraph H with tree duality and its set graph.

We now describe the *arc-consistency procedure* studied already in [11]. At the beginning, each vertex v of an input graph G is assigned the set $\ell_0(v)$ of all the vertices of the target graph H and the set assigned to v after i steps of the algorithm is denoted by $\ell_i(v)$. At the i -th step, a vertex $w \in V(H)$ is removed from the set of v if G contains an arc vv' such that H does not contain an arc ww' for any $w' \in \ell_{i-1}(v')$ or G contains an arc $v'v$ such that H does not contain an arc $w'w$ for any $w' \in \ell_{i-1}(v')$. We say that such an arc vv' is *violated* at the i -th step. The procedure terminates when there are no violated arcs. The number of steps never exceeds $|G| \cdot |H|$. The running of the procedure is linear in $|G| + ||G||$ when H is fixed (and when the assignments ℓ_i at each step are implicitly represented).

If there is a vertex v whose final set is empty, then G is not homomorphic to H . Otherwise, the mapping $h : V(G) \rightarrow V(2^H)$ that maps each v to its final set is a homomorphism from G to 2^H . If H has tree duality, then 2^H is homomorphic to H and thus G is homomorphic to H .

4 Target Graphs with Tree Duality

We show that if H has tree duality, then $\lim_{k \rightarrow \infty} \rho_k(H) = 1$, and we design an algorithm that either finds a good mapping from G to H or detects a subgraph of G of bounded size not homomorphic to H . Note that even if H has tree duality, the problem to maximize the number of satisfied constraints can be hard. For instance the problem whether G is homomorphic to the digraph consisting of a

single arc can be solved in polynomial time, but the maximization problem is hard: for an undirected graph G_0 , let G be the digraph obtained by replacing each edge with a bigon. The maximum number of arcs that can be preserved by a mapping from G to the arc is equal to the size of the maximum cut of G_0 .

Theorem 2. *If H is a digraph that has tree duality, then the following holds:*

$$\lim_{k \rightarrow \infty} \rho_k(H) = 1.$$

Moreover, there is an algorithm that for an input graph G and $\varepsilon > 0$ either finds a mapping $h : V(G) \rightarrow V(H)$ preserving at least $(1 - \varepsilon) \cdot \|G\|$ arcs or finds a subgraph of G of size at most $|H|^{\lceil 2|H|/\varepsilon \rceil}$ not homomorphic to H . The running time of the algorithm is linear in $|G| + \|G\|$ if the target graph H is fixed.

Proof. We first describe the algorithm. The algorithm runs the arc-consistency procedure for the first $\lceil 2|H|/\varepsilon \rceil$ steps and constructs the corresponding assignments ℓ_i . It then distinguishes two cases. The first case is that there exists a vertex v with $\ell_i(v) = \emptyset$. Let i be the smallest index with this property. For every $w \in V(H)$, there exists a step of the algorithm when w was removed from the set of v because an incident edge was violated. For $w \in V(H)$, consider such an edge $v_w v$ and the corresponding step i_w . Note that $i_w < i$. Now, for every $w' \in V(H)$ missing in $\ell_{i_w}(v_w)$, consider the step when w' was removed from the sets of v_w . Note that $\ell_{i_w}(v_w) \neq \emptyset$ by the choice of i . We obtain new sets of arcs that were violated before the i_w -th step and that caused vertices w' to be removed from the set of v_w . Continue in this way unless the sets assigned to the vertices of G are equal to $V(H)$. This terminates because the numbers i_w of steps decrease. Since $i \leq \lceil 2|H|/\varepsilon \rceil$, the number of considered violated arcs does not exceed:

$$|H| + |H|(|H| - 1) + |H|(|H| - 1)^2 + \dots + |H|(|H| - 1)^{\lceil 2|H|/\varepsilon \rceil - 1} \leq |H|^{\lceil 2|H|/\varepsilon \rceil}.$$

This set of arcs contains a subgraph of G that is not homomorphic to H .

The other case is that $\ell_i(v) \neq \emptyset$ for all $v \in V(G)$ and i . Let E_i be the set of the arcs violated at the i -th step. Since each edge is violated at most $2|H|$ times (at each step when the edge is violated, the size of the set of one of its end-vertices decreases), the sum $|E_1| + \dots + |E_{\lceil 2|H|/\varepsilon \rceil}|$ is bounded by $2|H| \cdot \|G\|$. Hence, there exists $i = 1, \dots, \lceil 2|H|/\varepsilon \rceil$ such that $|E_i| \leq \varepsilon \|G\|$. Consider a mapping $h' : V(G) \rightarrow V(2^H)$ defined as $h'(v) := \ell_{i-1}(v)$; h' preserves all arcs possibly except for those of E_i . Since 2^H is homomorphic to H , there is a homomorphism $h : V(G) \rightarrow V(H)$ that preserves at least $(1 - \varepsilon)\|G\|$ arcs. The bound on the running time of our algorithm follows from the discussions in Section 3.

Since the algorithm finds for a $|H|^{\lceil 2|H|/\varepsilon \rceil}$ -consistent G a mapping preserving at least $(1 - \varepsilon)\|G\|$ arcs, $\rho_{|H|^{\lceil 2|H|/\varepsilon \rceil}}(H) \geq 1 - \varepsilon$ and $\lim_{k \rightarrow \infty} \rho_k(H) = 1$.

Since $\rho_{2^k}^v(H) \geq \rho_k(H)$ and $\rho(H)$ is non-decreasing in k , Theorem 2 implies:

Corollary 3. *If H is a digraph that has tree duality, then the following holds:*

$$\lim_{k \rightarrow \infty} \rho_k^v(H) = 1.$$

Moreover, there is an algorithm that for an input graph G and $\varepsilon > 0$ either finds a mapping $h : V(G) \rightarrow V(H)$ preserving at least $(1 - \varepsilon) \cdot ||G||$ arcs or finds a subgraph of G of order at most $1 + |H|^{\lceil 2|H|/\varepsilon \rceil}$ not homomorphic to H . The running time of the algorithm is linear in $|G| + ||G||$ if H is fixed.

5 Directions for Future Research

The main interest in locally consistent CSPs comes from the question how much it helps that the input is locally consistent. This is reflected by the behavior of $\rho_k(H)$ as a function of k . In case that H contains a loop or a directed cycle, the assumption on local consistency does not help at all. On the other hand, if H has tree duality, this assumption helps a lot. We were not able to settle the case where H is acyclic but does not have tree duality. In Figure 2, there are the three smallest digraphs H for which we cannot compute the limit $\lim_{k \rightarrow \infty} \rho_k(H)$.

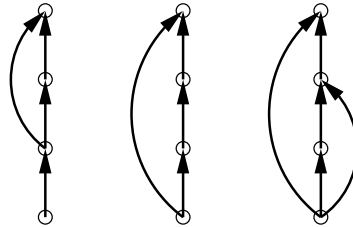


Fig. 2. The three smallest digraphs H for which $\lim_{k \rightarrow \infty} \rho_k(H)$ is unknown.

In [6, 7, 18], the authors also addressed the weighted versions of the problems. Let us mention that all our results, in particular Theorems 1 and 2, Corollaries 1, 2 and 3, hold for the weighted versions of the problems, too. The reader is welcome to check him/her/itself that the proofs translate to this setting.

The ultimate goal is to settle the behavior of locally consistent CSPs with more types of constraints and with constraints of arbitrary arity. The approach to CSPs for binary constraints based on graphs with tree duality applies to all constraint languages that admit a set function, even if the constraint language contains several constraint types. Note that this class of computational problems contains many previously known tractable families of problems including Horn, constant, and ACI problems [3]. However, we have little knowledge of the behavior for constraint languages without tree duality.

Acknowledgement

The authors are indebted to Zdeněk Dvořák for discussions on relative density of graphs and to Pavol Hell for his comments on tree duality.

References

1. A. Bulatov, A. Krokhin, P. Jeavons: The Complexity of Maximal Constraint Languages. In: Proc. 33rd Symp. on Theory of Computation, STOC (2001) 667–674.
2. S. Cook, D. Mitchell: Finding Hard Instances of the Satisfiability Problem: A Survey. In: Satisfiability Problem: Theory and Applications. DIMACS Series in DMTCS Vol. 35 AMS (1997).
3. V. Dalmau, J. Pearson: Closure Functions and Width 1 Problems. In: Proc. 5th International Conferences on Principles and Practice of Constraint Programming, CP, LNCS 1713, Springer Berlin (1999) 159–173.
4. R. Dechter, P. van Beek: Local and Global Relational Consistency. *Theor. Comput. Sci.* 173 (1997) 283–308.
5. Z. Dvořák: personal communication.
6. Z. Dvořák, D. Král', O. Pangrác: Locally Consistent Constraint Satisfaction Problems. In: Proc. 31st International Colloquium on Automata, Languages and Programming, ICALP, LNCS 3142, Springer-Verlag Berlin (2004) 469–480.
7. Z. Dvořák, D. Král', O. Pangrác: Locally Consistent Constraint Satisfaction Problems, to appear in *Theor. Comput. Sci.*
8. D. Eppstein: Improved Algorithms for 3-coloring, 3-edge-coloring and Constraint Satisfaction. In: Proc. 12th ACM-SIAM Symposium on Discrete Algorithms, SODA (2001) 329–337.
9. T. Feder, R. Motwani: Worst-case Time Bounds for Coloring and Satisfiability Problems. *J. Algorithms* 45(2) (2002) 192–201.
10. T. Feder, M. Vardi: Monotone monadic SNP and constraint satisfaction. In: Proc. 25th Symposium on the Theory of Computation, STOC (1993) 612–622.
11. E. C. Freuder: A sufficient condition for backtrack-free search. *J. ACM* 29 (1982) 24–32.
12. P. Hell, J. Nešetřil: *Graphs and homomorphisms*. Oxford University Press (2004).
13. P. Hell, J. Nešetřil, X. Zhu: Duality and polynomial testing of tree homomorphisms. *Trans. Amer. Math. Soc.* 348(4) (1996) 1281–1297.
14. M. A. Huang, K. Lieberherr: Implications of Forbidden Structures for Extremal Algorithmic Problems. *Theor. Comput. Sci.* 40 (1985) 195–210.
15. S. Janson, T. Łuczak, A. Ruciński: *Random Graphs*. Wiley, New York (2000).
16. S. Jukna: *Extremal Combinatorics with Applications in Computer Science*. Springer, Heidelberg (2001).
17. D. Král': Locally Satisfiable Formulas. In: Proc. 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). SIAM (2004) 323–332.
18. D. Král', O. Pangrác: An Asymptotically Optimal Linear-Time Algorithm for Locally Consistent Constraint Satisfaction Problems, submitted.
19. K. Lieberherr, E. Specker: Complexity of Partial Satisfaction. *J. of the ACM*, 28(2) (1981) 411–422.
20. K. Lieberherr, E. Specker: Complexity of Partial Satisfaction II. Technical Report 293, Dept. of EECS, Princeton University (1982).
21. Z. Usiskin: Max-min Probabilities in the Voting Paradox. *Ann. Math. Stat.* 35 (1963) 857–862.
22. L. Trevisan: On Local versus Global Satisfiability, *SIAM J. Discrete Math.*, 17(4) (2004), 541–547. A preliminary version available as ECCO report TR97-12.
23. G. J. Woeginger: Exact Algorithms for NP-hard Problems: A Survey. In: Proc. Worksh. Comb. Opt.—Eureka, You Shrink. LNCS 2570, Springer (2003) 185–207.
24. M. Yannakakis: On the Approximation of Maximum Satisfiability. *J. Algorithms* 17 (1994) 475–502.