A faster polynomial-space algorithm for Max 2-CSP

Edwards, Keith

Published in:
Journal of Computer and System Sciences

DOI:
10.1016/j.jcss.2015.11.013

Publication date:
2016

Document Version
Peer reviewed version

A faster polynomial-space algorithm for Max 2-CSP

Keith Edwards
School of Computing
University of Dundee
Dundee, DD1 4HN
U.K.
kjedwards@dundee.ac.uk
Tel: +44 1382 384463

November 26, 2014

Abstract

We give an algorithm for Max 2-CSP which runs in time $O^\star(r^{\frac{m}{5.555}})$ and uses polynomial space. This compares with the previous fastest published algorithm of Scott and Sorkin [15], which runs in time $O^\star(r^{\frac{19m}{100}}) = O^\star(r^{\frac{m}{5.263}})$.

The algorithm uses the deletion-reduction depth of a graph; we also consider some of the properties of this parameter, and in particular derive a lower bound on the parameter for cubic graphs.

Keywords: Max 2-CSP; deletion-reduction depth; deletion depth.

1 Introduction

There has been a long sequence of exponential time algorithms for problems contained in Max 2-CSP, particularly for Max Cut and Max 2-Sat, and more recently Max 2-CSP itself.

Kulikov and Fedin [10] gave an $O^\star(2^{m/4})$ algorithm for Max Cut, and this was improved to $O^\star(2^{m/5})$ for general Max 2-CSP by Scott and Sorkin [13]. This was further improved by Scott and Sorkin [14, 15] and Gaspers and Sorkin [5], who describe an algorithm with time complexity $O^\star(r^{19m/100}) = O^\star(r^{m/5.263})$ (where r is the domain size for each variable). This has remained the best published polynomial-space algorithm for Max 2-CSP (and Max Cut) parameterized by m (the number of edges in the graph), although there have been subsequent improvements, described below, in the complexity expressed in terms of the average degree.

For Max 2-CSP parameterized by the average degree d and number of vertices n, Scott and Sorkin [15] give an algorithm which requires time at most $O^\star(r^{(1-\frac{2}{d+1})n})$. This is improved to $O^\star(r^{(1-\frac{3}{d+1})n})$ by Golovnev and Kutzkov [7] and to $O^\star(r^{(1-\frac{3.25}{d+1}+O(1/d^3))n})$ in [3]. Golovnev and Kutzkov also give an algorithm which for very large average degree d has a smaller exponent of the form $(1 - O(\frac{\ln d}{d}))n$.

We describe an algorithm which runs in time $O^\star(r^{9m/50+o(m)}) = O^\star(r^{m/5.555})$ and polynomial space, or, in terms of average degree, in time $O^\star(r^{(1-\frac{3.4}{d+1}+O(1/d^3))n})$.

A very recent preprint by Gaspers and Sorkin [6] also gives an algorithm for Max 2-CSP running in time $O^\star(r^{9m/50+o(m)})$ and polynomial space, or, in terms of average degree, in time $O^\star(r^{(1-\frac{3.3}{d+1})n})$. It is based on similar ideas to the ones used in this paper, but the detailed method is very different. Among the distinctive features of our approach are the following: (1) We clearly separate the dependency on the underlying graph from the solution of the CSP instance itself, by using the notions of deletion-reduction depth and deletion-reduction tree, which depend only on the graph; (2) Our uniform treatment of all degrees greater than 3, and analysis of the function g_α (see Section 2.3), give a better upper bound in terms of average degree; (3) We use a general reduction theorem to extend the results from cubic graphs to general graphs, avoiding the need for any new induction and setting the result in a broader context; (4) Our analysis of the relationship between deletion-reduction depth and deletion depth allows us (a) to show that essentially the same upper bound can be obtained using only branching on deleted vertices, without any other reductions, and (b) to give a lower bound on the deletion-reduction depth in the cubic case and therefore indicate the probable limits of this approach.

The fastest *exponential-space* algorithm remains that of Williams [17], which solves constraint satisfaction problems in time $O^\star(r^{\omega n/3})$, where ω is the matrix multiplication constant ($\omega \approx 2.373$).

2 Preliminaries

All graphs $G = (V, E)$ are considered to be simple. We use standard graph terminology and notation, hence $n = |V|$ and $m = |E|$, and the minimum degree of G is $\delta(G)$. We describe the efficiency of an exponential-time algorithm using the standard $O^\star$ notation, which suppresses polynomial factors in any parameters.

If $X \subseteq V(G)$, we write $G - X$ to mean the graph formed by deleting the vertices in X (and incident edges) from G. In the case that $X = \{v\}$, we will write $G - v$ rather than $G - \{v\}$.

2.1 Max 2-CSP

An instance (G, S) of the maximum 2-constraint satisfaction problem (Max 2-CSP) consists of a simple graph $G = (V, E)$, called the *constraint graph*, a set of colours $[r] = \{0, 1, \ldots, r-1\}$, for some $r \geq 2$, a constant $S_\emptyset$ and, for each edge and vertex of G, a score function, where the score function of an edge uv takes the form $S_{uv} : [r]^2 \to \mathbb{R}$, and the score function of a vertex v is of the form $S_v : [r] \to \mathbb{R}$. Note that only one score is defined for a given colouring of each edge uv, so for colours $c_1, c_2 \in [r]$ we consider $S_{uv}(c_1, c_2)$ and $S_{vu}(c_2, c_1)$ to be equivalent names for the same score.

Any colouring ψ of the vertices of G using the set of colours $[r]$ induces a *cost* which is the sum of the vertex and edge functions plus $S_\emptyset$:

$$S(\psi) = S_\emptyset + \sum_{v \in V} S_v(\psi(v)) + \sum_{uv \in E} S_{uv}(\psi(u), \psi(v))$$

A *candidate solution*, or more simply, a *solution* of a Max 2-CSP instance is any function $\psi : V \to [r]$ which assigns a colour to each vertex of G. An *optimal solution* is a solution which maximizes $S(\psi)$, and the goal of the Max 2-CSP problem is to find an optimal solution.

A number of problems such as Maximum Cut, Maximum Directed Cut, Maximum Independent Set, Minimum Vertex Cover, and Maximum 2-Sat can be modelled as a Max 2-CSP instance.

2.2 Reduction

We will need below the concept of series-parallel reductions. Let G be a graph, and consider the following four operations on G:

R0 Delete an isolated vertex of G.

R1 Delete a vertex of degree 1 (and its incident edge).

R2n Let v be a vertex of degree 2 with non-adjacent neighbours x and y; delete v (and edges vx, vy) and add an edge between x and y.

R2a Let v be a vertex of degree 2 with adjacent neighbours; delete v (and incident edges).

Let $r(G)$ be the graph obtained from G by applying operations R0, R1, R2n, R2a above repeatedly until none is possible (because the graph has minimum degree at least 3 or is empty). It follows from Theorem 4.1, proved by Kneis *et al.* [8], that $r(G)$ is well-defined, i.e., that the resulting graph does not depend on the order of operations. It also follows from the definition of the series-parallel property that G is series-parallel if and only if $r(G)$ is empty. The reductions R0 R1, R2n, R2a have been used by many authors.

2.3 Definition and properties of function g_α

We now define the important function g_α introduced in [3] and restate some of its properties.

Definition of functions g_α and g'_α

For any $n \geq 2$, and α with $0 \leq \alpha \leq 1$, define the function $g_\alpha(n)$ by setting $g_\alpha(2) = 0$, $g_\alpha(3) = \alpha$, and for any $n \geq 4$,

$$ng_\alpha(n) = (n-2)g_\alpha(n-1) + g_\alpha(n-2) + 1.$$

We extend g_α to all real numbers at least 2 by linear interpolation, i.e. if $r = n+x$, where $n \geq 2$ is an integer, and $0 \leq x \leq 1$, then we set $g_\alpha(r) = (1-x)g_\alpha(n) + xg_\alpha(n+1)$.

Also, for any $n \geq 3$, define $g'_\alpha(n) = g_\alpha(n) - g_\alpha(n-1)$.

Properties of g_α

We now give the properties of g_α which we will use later. The first three are proved in [3], the fourth is proved for $\alpha = \frac{1}{4}$ in [2], and this proof carries over to $\alpha \in [\frac{1}{6}, \frac{3}{10}]$.

G1. For any integer $n \geq 2$,

$$g_\alpha(n) = (4-3\alpha)\frac{A(n)}{n!} + (2-3\alpha)\frac{(-1)^n}{n!} - (3-3\alpha)$$

where $A(n)$ is the alternating factorial function given by

$$A(n) = n! - (n-1)! + \ldots - (-1)^n \cdot 1!.$$

G2. For all real $d \geq 2$,

$$g_\alpha(d) = 1 - \frac{4-3\alpha}{d+1} + O(1/d^3).$$

G3. If $1/6 \leq \alpha \leq 3/10$, $g_\alpha(n)/n$ is strictly decreasing for $n \geq 5$.

G4. If G is a graph with average degree d, and either G is connected and $d \geq 2$, or $d \geq 5$, then for $\alpha \in [\frac{1}{6}, \frac{3}{10}]$,

$$\sum_{v \in V(r(G))} g_\alpha(d_{r(G)}(v)) \leq g_\alpha(d)|V(G)|.$$

In this paper we use the function $g_{1/5}$. The first few values of this function are given below:

d	2	3	4	5	6	7	8
$g_{1/5}(d)$	0	$\frac{1}{5}$	$\frac{7}{20}$	$\frac{9}{20}$	$\frac{21}{40}$	$\frac{163}{280}$	$\frac{281}{448}$

Definitions of functions f_α, f_α^-

We now define the functions which will be used as upper bounds. For any graph G with minimum degree at least 2, define $f_\alpha(G)$ by:

$$f_\alpha(G) = \sum_{v \in V(G)} g_\alpha(d_G(v)).$$

Although $f_\alpha(G)$ is the main upper bound of interest, the inductive proof relies crucially on a small negative term, which depends on the minimum degree of each component. If G has minimum degree at least 3, with connected components $G_1, \ldots, G_k$, set

$$f_\alpha^-(G) = f_\alpha(G) - \sum_{i=1}^{k} g'_\alpha(\delta(G_i)) \quad = \sum_{i=1}^{k} f_\alpha^-(G_i).$$

2.4 The general reduction theorem

In [3] we prove the following general reduction theorem, which we will use later. We need one further definition.

Definition of basic graph

We will say a connected graph of minimum degree at least 3 is basic if either G is cubic or $r(G - v)$ is empty for some $v \in V$. A graph is basic if every component is basic.

Theorem 2.1 ([3]) *Let $G = (V, E)$ be a graph with n vertices and minimum degree $\delta \geq 3$. Then there is a set $X_G \subseteq V$ such that (i) $r(G - X_G)$ is non-empty and basic, and (ii) $|X_G| + f_\alpha^-(r(G - X_G)) \leq f_\alpha^-(G)$ for any α, $1/6 \leq \alpha \leq 3/10$.*

3 Deletion-Reduction depth

Let G be a graph. By a deletion-reduction operation we mean an operation which replaces G by $r(G - v)$, where v is any vertex of G. We define the deletion-reduction depth of a graph to be the least number of times that applying a deletion-reduction operation on each component of the graph, starting from $r(G)$, can result in an empty graph.

3.1 Definition of deletion-reduction depth

The deletion-reduction depth $\mathrm{drd}(G)$ is the least integer k such that there exists a sequence $G_0 = r(G), G_1, \ldots, G_k$, with G_k the empty graph, and a sequence $V_0, \ldots, V_{k-1}$ of vertex sets, such that for each $i = 1, \ldots, k$,

(a) V_{i-1} is a subset of $V(G_{i-1})$ containing at most one vertex from each component of G_{i-1};

(b) $G_i = r(G_{i-1} - V_{i-1})$.

3.2 Deletion-reduction tree

The notion of deletion-reduction depth naturally suggests a corresponding tree of reductions and deletions, which turns out to be just the reduction tree defined by Scott and Sorkin [15]. (Note we restrict ourselves here to the case where G is connected, for the case where G is not connected we would obtain a deletion-reduction forest consisting of a tree for each component of G.)

Definition of deletion-reduction sequence

Let G be a graph with n vertices. A deletion-reduction sequence on G is a sequence of operations $\phi_1, \ldots, \phi_n$, where each operation is either one of R0, R1, R2n, R2a or is the deletion of a single vertex, such that if $G^{(0)} = G$ and $G^{(i)} = \phi_i(G^{(i-1)})$ for $i = 1, \ldots, n$, then $G^{(n)}$ is empty. Note that each operation removes exactly one vertex from the graph.

Definition of deletion-reduction tree

Let G be a graph with n vertices, and let $\phi_1, \ldots, \phi_n$ be a deletion-reduction sequence on G. Then the deletion-reduction tree T for this sequence is a rooted tree, as follows:

The vertex set of T is the same as that of G. The root of T is the vertex removed by the first operation ϕ_1. Now consider any operation ϕ_i, and suppose that ϕ_i removes a vertex v. Let C be the component (of $G^{(i-1)}$) containing v; the operation will have no affect on any other component. Let $C_1, \ldots, C_c$ be the components of $\phi_i(C)$; here the number of components, c, will be 0 in the case of an R0 reduction, 1 for an R1, R2n or R2a reduction, and at least 1 for a deletion. Let $v_1, \ldots, v_c$ be the vertices removed

first from $C_1, \ldots, C_c$ respectively, and make these the children of v in T. Finally we label each vertex in T with the type of the operation which removes it from G, i.e., one of R0, R1, R2n, R2a for a reduction, or D for a deletion (these labels are usually redundant because they are implicit in the structure, but are included for completeness to allow for an arbitrary deletion-reduction sequence possibly including deletions of vertices of degree less than 3).

Remarks: (1) Thus the root of the tree is the first node to be reduced or deleted, its children are the vertices first reduced/deleted from each resulting component, and so on.

(2) The set of vertices in the subtree rooted at any vertex v is exactly the vertex set of the component C containing v when it is removed from the graph.

Relationship to deletion-reduction depth

Consider a graph G with deletion-reduction depth k. Let $V_0, \ldots, V_{k-1}$ be the sets of vertices deleted at each stage, and let $G_0 = r(G), G_1, \ldots, G_k$, with G_k the empty graph, be the corresponding sequence of graphs.

Let r_i be a sequence of series-parallel reductions which forms G_i from $G_{i-1} - V_{i-1}$ (for $i = 0$, r_0 forms G_0 from G), and d_i be a sequence of deletions (one vertex at a time) of the vertices in V_i. Then concatenating the sequences $r_0, d_0, r_1, d_1, \ldots, d_{k-1}, r_k$ gives a deletion-reduction sequence for G. Furthermore, it is easy to see that in the corresponding deletion-reduction tree, the number of deletions on any path from the root to a leaf is at most $k = \mathrm{drd}(G)$.

3.3 Deletion-reduction depth and Max 2-CSP

As mentioned above the concepts of deletion-reduction tree and deletion-reduction depth are not really new; a deletion-reduction tree is just the reduction tree defined by Scott and Sorkin [15] in their description of their algorithm B for Max 2-CSP. They also define the III-reduction depth of a reduction tree as the largest number of III-reductions (i.e., deletions) on any path from the root to a leaf of the tree, and so for an optimal tree this is the deletion-reduction depth. (Note that they refer to our R0, R1 and R2 reductions as 0-, I- and II-reductions respectively.)

The reduction tree is constructed, as above, in part B.1 of Scott and Sorkin's algorithm B. Parts B.2 and B.3 each take as their input an instance (G, S) of Max 2-CSP together with a reduction tree T for the graph G, and perform a depth-first (pre-order) traversal of the tree. Part B.2 outputs the optimum cost and part B.3 a colouring realising this optimum cost.

They prove the following lemma (Claims 9 and 10 of [15]):

Lemma 3.1 ([15]) *For a CSP instance (G, S) where G has n nodes and has a reduction tree with III-reduction depth d, Algorithms B.2 and B.3 run in time $O(nr^{3+d})$ and in linear space.* □

It follows immediately from this lemma that Max 2-CSP can be solved in time $O^\star(r^{\mathrm{drd}(G)})$, and polynomial space, on a graph G.

3.4 Cubic Graphs

We first show that for a cubic graph G on n vertices, the deletion-reduction depth $\mathrm{drd}(G)$ is at most $(\frac{1}{5} + o(1))n$. We make use of a result of Monien and Preis [11] which bounds the size of a cut dividing a cubic graph into two (almost) equal parts.

The bisection width $\mathrm{bw}(G)$ of a graph G is defined as follows (from [11]). Let $\pi : V \to \{0, 1\}$ be a bisection of G, dividing the vertices into parts $A = V_0$ and $B = V_1$. A balanced bisection is one in which the number of vertices in the parts differ by at most 1. Let $\mathrm{cut}(\pi) := |\{vw \in E \,|\, \pi(v) \neq \pi(w)\}|$ be the cut size of π. The bisection width of a graph G is $\mathrm{bw}(G) := \min\{\mathrm{cut}(\pi); \pi \text{ is a balanced bisection of } G\}$.

Theorem 3.2 (Monien, Preis [11]) *For any $\varepsilon > 0$ there is a value $n(\varepsilon)$ such that the bisection width of any 3-regular graph $G = (V, E)$ with $|V| > n(\varepsilon)$ is at most $(\frac{1}{6} + \varepsilon)|V|$.* □

It is noted in [4] that Theorem 3.2 is constructive and can be turned into polynomial time algorithm which constructs a cut of the required size.

We first give a very rough sketch of the main idea. Given a cubic graph, we can find a cut of size roughly $n/6$. We apply deletion-reduction to one endpoint of each edge in the cut, with $n/12$ of the endpoints in each part of the partition. Each deletion-reduction results in 3 vertices being removed in the same part of the partition, and 1 in the other part. In total roughly $n/3$ vertices are removed from each part, and the two parts become disconnected. Hence we obtain a graph with no component larger than $n/6$. By induction this graph has deletion-reduction depth at most $n/30$ approximately, giving a total deletion-reduction depth of roughly $n/6 + n/30 = n/5$. To make this argument precise, we need to take account of a number of complications, in particular that a vertex may be incident with more than one edge of the cut, and that a cut with fewer than $n/6$ edges will result in larger parts in the final partition, so it is really the balance between the number of deletion-reduction operations and the size of the resulting parts which is important. We give details below.

We start with a lemma which concerns the use of deletion-reduction operations on the endpoints of the cut to disconnect the graph.

Lemma 3.3 *Let G be a cubic graph with n vertices, and with bisection width $\mathrm{bw}(G)$. Then there is a sequence of k deletion-reduction operations which will leave a graph with no component of more than c vertices, such that*

$$\lambda k + c \leq \frac{n}{2} + (\lambda - 2)\mathrm{bw}(G) + 1$$

for any $\lambda \geq 3$.

Proof. First let $\pi = (A, B)$ be any partition of the vertices of a cubic graph into sets A and B, and let $\text{cut}(\pi)$ be the number of edges between parts A and B. Also let D be a set disjoint from $A \cup B$ (below, D will be the set of vertices deleted from the graph).

Consider the following operation:

> **Cut-decreasing step**: (i) If some vertex v is incident with two or three edges in the cut, then move v to the other part of the partition; otherwise (ii) choose an edge of the cut, and let v be the endpoint of this edge belonging to the larger of A and B (choose arbitrarily if $|A| = |B|$). Perform a deletion-reduction operation on v, i.e. delete v from the graph and reduce the resulting graph to form $r(G - v)$. Add v to the set D.

To analyse the effect of the cut-decreasing step, it is helpful to introduce a function $\overline{\max}(x, y)$ which differs from $\max(x, y)$ by at most one, but has a useful property (property M5 below) which the maximum function itself lacks. For any real numbers x, y, define the function $\overline{\max}(x, y)$ by

$$\overline{\max}(x, y) = \min(\max(x, y - 1), \max(x - 1, y)).$$

It is easy (see the Appendix) to verify that $\overline{\max}$ has the following properties:

M1. $\overline{\max}$ is monotone non-decreasing in each argument;

M2. $\max(x, y) \geq \overline{\max}(x, y) \geq \max(x, y) - 1$;

M3. If $|x - y| \leq 1$, $\overline{\max}(x, y) = \min(x, y)$;

M4. $\overline{\max}(x - 1, y + 1) \leq \overline{\max}(x, y) + 1$;

M5. If $x' \leq \max(x, y) - 3$, and $y' \leq \min(x, y) - 1$, then $\overline{\max}(x', y') \leq \overline{\max}(x, y) - 2$.

Now suppose that $\lambda \geq 3$, and consider the following quantity:

$$\lambda|D| + \overline{\max}(|A|, |B|) + (\lambda - 2)\text{cut}(\pi).$$

We show that the cut-decreasing step does not increase this quantity. The first operation (i), i.e. moving a vertex from A to B or vice versa, increases $\overline{\max}(|A|, |B|)$ by at most 1 (by property M4 above), and decreases $\text{cut}(\pi)$ by at least 1, while D is not changed. Thus since $\lambda - 2 \geq 1$, the quantity $\lambda|D| + \overline{\max}(|A|, |B|) + (\lambda - 2)\text{cut}(\pi)$ does not increase.

The second operation (ii) is a deletion-reduction of a vertex in the larger part of the partition. Since the deleted vertex v is incident with exactly one edge of the cut, deleting v will leave two vertices of degree two in this larger part and one in the other part. These will be removed by reduction. Some further reductions may take place, but it is easy to see that these cannot increase any of $|A|$, $|B|$ or $\text{cut}(\pi)$. This is because the only reduction which adds a new edge xy is one of type R2n. In that case the vertex v reduced is already adjacent to both x and y, and the reduction removes both of the edges vx and vy and replaces them with the edge xy. But if xy is in the cut, then so was one of vx and vy, hence $\text{cut}(\pi)$ does not increase.

Thus in case (ii), the size of the larger of A and B reduces by at least 3 and the smaller by at least 1, so by property M5 of the function $\overline{\max}$, the value of $\overline{\max}(|A|,|B|)$ reduces by at least 2. Also $\mathrm{cut}(\pi)$ reduces by at least 1, and $|D|$ increases by 1. Hence $\lambda|D| + \overline{\max}(|A|,|B|) + (\lambda-2)\mathrm{cut}(\pi)$ does not increase.

Now let $\pi_I = (A_I, B_I)$ be a balanced bisection of size $\mathrm{bw}(G)$ which divides the graph G into sets A_I and B_I. Also let $D_I = \emptyset$.

Starting with the partition π_I, we iterate the cut-decreasing step as long as possible, i.e., as long as the cut is non-empty. Let the final partition be $\pi_F = (A_F, B_F)$, and let D_F be the final set of deleted vertices.

Then since the cut-decreasing step does not increase $\lambda|D| + \overline{\max}(|A|,|B|) + (\lambda-2)\mathrm{cut}(\pi)$, we have

$$\lambda|D_F| + \overline{\max}(|A_F|,|B_F|) + (\lambda-2)\mathrm{cut}(\pi_F) \leq \lambda|D_I| + \overline{\max}(|A_I|,|B_I|) + (\lambda-2)\mathrm{cut}(\pi_I).$$

Now $\overline{\max}(|A_I|,|B_I|) \leq \min(|A_I|,|B_I|) \leq \frac{n}{2}$ by property M3, $|D_I| = 0$ and $\mathrm{cut}(\pi_I) = \mathrm{bw}(G)$ so

$$\lambda|D_I| + \overline{\max}(|A_I|,|B_I|) + (\lambda-2)\mathrm{cut}(\pi_I) \leq \frac{n}{2} + (\lambda-2)\mathrm{bw}(G).$$

Also, we have $\mathrm{cut}(\pi_F) = 0$ and $k = |D_F|$. Thus

$$\lambda|D_F| + \overline{\max}(|A_F|,|B_F|) + (\lambda-2)\mathrm{cut}(\pi_F) = \lambda k + \overline{\max}(|A_F|,|B_F|),$$

and so

$$\lambda k + \overline{\max}(|A_F|,|B_F|) \leq \frac{n}{2} + (\lambda-2)\mathrm{bw}(G).$$

Also it is clear that the maximum component size c in the final graph satisfies $c \leq \max(|A_F|,|B_F|) \leq \overline{\max}(|A_F|,|B_F|) + 1$, so

$$\lambda k + c \leq \frac{n}{2} + (\lambda-2)\mathrm{bw}(G) + 1,$$

as required. □

Lemma 3.4 *Let G be a cubic graph with n vertices. Then*

$$\mathrm{drd}(G) \leq \left(\frac{1}{5} + o(1)\right) n.$$

Proof. We will show that for any $\mu > 0$, there is a constant C_μ such that $\mathrm{drd}(G) \leq \frac{n}{5} + \mu n + C_\mu$ for any cubic graph G on n vertices. The result follows immediately from this.

Let $\varepsilon = \min(\frac{\mu}{12}, \frac{1}{6})$ and $C_\mu = \max\{n(\varepsilon), 12, \lceil \frac{6}{\mu} \rceil\}$ (here $n(\varepsilon)$ is the constant provided by Theorem 3.2 of Monien and Preis [11]). We proceed by induction on n. First if $n \leq C_\mu$, then it is immediate that $\mathrm{drd}(G) \leq C_\mu$, so the result holds. So suppose that $n \geq C_\mu$.

Then since $n \geq n(\varepsilon)$, we have $\mathrm{bw}(G) \leq (\frac{1}{6} + \varepsilon)n$. Since $\varepsilon \leq \frac{1}{6}$, we have in particular $\mathrm{bw}(G) \leq \frac{n}{3}$.

By Lemma 3.3 there is a sequence of k deletion-reduction operations which will leave a graph G' with no component of more than c vertices, such that

$$\lambda k + c \leq \frac{n}{2} + (\lambda - 2)\mathrm{bw}(G) + 1$$

for any $\lambda \geq 3$. Taking first $\lambda = 3$, we obtain

$$c \leq \frac{n}{2} + \frac{n}{3} + 1 = \frac{5}{6}n + 1 \leq \frac{5}{6}n + \frac{1}{12}n = \frac{11}{12}n$$

since $n \geq 12$. Then taking $\lambda = 5$, we obtain

$$5k + c \leq \frac{n}{2} + 3\mathrm{bw}(G) + 1 \leq \frac{n}{2} + 3\left(\frac{1}{6} + \varepsilon\right)n + 1 = n + 3\varepsilon n + 1$$

or

$$k + \frac{c}{5} \leq \frac{n}{5} + \frac{3\varepsilon n}{5} + \frac{1}{5}. \tag{1}$$

It is clear that $\mathrm{drd}(G) \leq k + \mathrm{drd}(G')$, and, if G' has components $C_1, \ldots, C_r$, then

$$\mathrm{drd}(G') = \max\{\mathrm{drd}(C_1), \ldots, \mathrm{drd}(C_r)\}.$$

Since each component C_i has at most c vertices, and $c < n$, then by the inductive hypothesis,

$$\mathrm{drd}(C_i) \leq \frac{c}{5} + \mu c + C_\mu$$

so

$$\mathrm{drd}(G) \leq k + \frac{c}{5} + \mu c + C_\mu.$$

Hence by (1)

$$\begin{aligned}\mathrm{drd}(G) &\leq \frac{n}{5} + \frac{3\varepsilon n}{5} + \frac{1}{5} + \mu c + C_\mu \\ &\leq \frac{n}{5} + \frac{3\mu n}{60} + \frac{\mu n}{30} + \mu\frac{11n}{12} + C_\mu \\ &= \frac{n}{5} + \mu n + C_\mu\end{aligned}$$

as required (here we also use $\varepsilon \leq \mu/12$, and $n \geq 6/\mu$, from which $1/5 \leq \mu n/30$). □

Remark We note that if the upper bound on $\mathrm{bw}(G)$ for cubic graphs were improved, for example to $\alpha n + o(n)$ with $\alpha < 1/6$, this would yield (by the same argument) a bound on $\mathrm{drd}(G)$ of $\beta n + o(n)$, where $\beta = \frac{2\alpha}{1+4\alpha}$.

Lemma 3.5 *Let $G = (V, E)$ be a graph with n vertices and minimum degree $\delta \geq 3$. Then*

$$\mathrm{drd}(G) \leq \sum_{v \in V(G)} g_{1/5}(d(v)) + o(n).$$

Proof. We will show that $\mathrm{drd}(G) \leq f^-_{1/5}(G) + o(n)$.

We first show that for any basic graph G, we have

$$\mathrm{drd}(G) \leq f^-_{1/5}(G) + o(n).$$

For a connected cubic graph G, this follows immediately from Lemma 3.4. If $r(G-v)$ is empty, then $\mathrm{drd}(r(G-v)) = 0$, and it follows from the result of Kneis *et al.*(see Theorem 4.1) or from Lemma 4.11, that $\mathrm{drd}(G) \leq 1$, so certainly $\mathrm{drd}(G) = o(n)$ as required. If G is basic with connected components $G_1, \ldots, G_k$, then

$$\begin{aligned}\mathrm{drd}(G) &= \max_i \mathrm{drd}(G_i) \\ &\leq \max_i \left(f^-_{1/5}(G_i) + o(n) \right) \\ &\leq f^-_{1/5}(G) + o(n).\end{aligned}$$

Now, by Theorem 2.1, for any graph G with minimum degree at least 3, there is a set X_G such that

$$|X_G| + f^-_{1/5}(r(G - X_G)) \leq f^-_{1/5}(G)$$

and $r(G-X_G)$ is basic and non-empty. From the construction of X_G (or from Lemma 4.11) it follows immediately that $\mathrm{drd}(G) \leq |X_G| + \mathrm{drd}(r(G - X_G))$, so we have

$$\begin{aligned}\mathrm{drd}(G) &\leq |X_G| + \mathrm{drd}(r(G - X_G)) \\ &\leq |X_G| + f^-_{1/5}(r(G - X_G)) + o(n) \quad \text{(since } r(G - X_G) \text{ is basic)} \\ &\leq f^-_{1/5}(G) + o(n),\end{aligned}$$

as required. □

Corollary 3.6 *Let G be a graph, and $r(G)$ be the reduced graph of G. Let n_d be the number of vertices of degree d in $r(G)$. Then*

$$\mathrm{drd}(G) \leq \sum_{v \in V(r(G))} g_{1/5}(d_{r(G)}(v)) + o(|V(G)|) = \sum_{d \geq 3} g_{1/5}(d) n_d + o(|V(G)|).$$

□

Lemma 3.7 *Let G be a graph with m edges. Then*

$$\mathrm{drd}(G) \leq \frac{9}{50} m + o(m).$$

Proof. We can assume that G is connected since the general result follows from the connected case. Recall (Property G3) that $g_{1/5}(d)/d$ is strictly decreasing for $d \geq 5$; it follows easily that $2g_{1/5}(d)/d$ attains its maximum value (of $9/50$) at $d = 5$. Let m' be the number of edges of $r(G)$, so that $m' \leq m$, and let n_d be the number of vertices of degree d in $r(G)$. Then from above we have

$$\begin{aligned}\mathrm{drd}(G) &\leq \sum_{d \geq 3} g_{1/5}(d) n_d + o(n) \\ &= \sum_{d \geq 3} \left(\frac{2g_{1/5}(d)}{d} \right) \left(\frac{dn_d}{2} \right) + o(n) \\ &\leq \left(\frac{2g_{1/5}(5)}{5} \right) \sum_{d \geq 3} \frac{dn_d}{2} + o(n) \\ &= \frac{9}{50} m' + o(n) \\ &\leq \frac{9}{50} m + o(m).\end{aligned}$$

□

For average degree, we obtain the following (by property G4 of g_α):

Theorem 3.8 *Let G be a graph of average degree $d \geq 2$, with n vertices and m edges. Then if G is connected, or $d \geq 5$,*

$$\mathrm{drd}(G) \leq (g_{1/5}(d) + o(1))n = (2g_{1/5}(d)/d + o(1))m = (1 - \tfrac{17/5}{d+1} + O(1/d^3))n.$$

□

The bound on deletion-reduction depth from Theorem 3.8 leads immediately to a faster algorithm for Max 2-CSP.

Theorem 3.9 *Let G be a graph, and $r(G)$ the reduced graph of G. Then an instance of Max 2-CSP on graph G can be solved in time*

$$O^\star(r^{\beta(G)}),$$

and polynomial space, where $\beta(G) = \sum_{v \in V(r(G))} g_{1/5}(d_{r(G)}(v)) + o(|V(r(G)|)$. □

In terms of edges, we obtain the following from Lemma 3.7:

Theorem 3.10 *Let G be a graph with m edges. Then an instance of Max 2-CSP on graph G can be solved in time*

$$O^\star(r^{\frac{9m}{50}+o(m)})$$

and polynomial space. □

Thus we obtain an algorithm which runs in time $O^\star(r^{\frac{m}{5.555}})$, compared with the previous fastest algorithm of Scott and Sorkin [15], which runs in time $O^\star(r^{\frac{19m}{100}}) = O^\star(r^{\frac{m}{5.263}})$.

We also have

Theorem 3.11 *Let G be a connected graph with n vertices and m edges, of average degree $d \geq 2$. Then an instance of Max 2-CSP on graph G can be solved in time*

$$O^\star(r^{(g_{1/5}(d)+o(1))n}) = O^\star(r^{(2g_{1/5}(d)/d+o(1))m}) = O^\star(r^{(1-\frac{17/5}{d+1}+O(1/d^3))n})$$

and polymonial space. □

This improves on the algorithm with complexity $O^\star(r^{(1-\frac{13/4}{d+1}+O(1/d^3))n})$ given in [3].

Extension to polynomial 2-CSP

A wider class of problems called polynomial 2-CSP is defined by Scott and Sorkin [16]. They show that several existing algorithms, including their Algorithm B from [15] which uses reduction trees, can be extended to deal with these problems. It is clear that the new algorithm above can be extended in the same way.

4 Deletion depth

As well as deletion-reduction depth defined above, it seems natural to also consider the simpler concept of deletion depth, in which series-parallel reductions are not used. Thus the deletion depth is the least number of times that deleting a vertex from each component of a graph leads to an empty graph.

4.1 Definition of deletion depth

The deletion depth $\mathrm{dd}(G)$ is the least integer k such that there exists a sequence $G_0 = G, G_1, \ldots, G_k$, with G_k the empty graph, and a sequence $V_0, \ldots, V_{k-1}$ of vertex sets, such that for each $i = 1, \ldots, k$,

(a) V_{i-1} is a subset of $V(G_{i-1})$ containing at most one vertex from each component of G_{i-1};

(b) $G_i = G_{i-1} - V_{i-1}$.

Note that whereas in the definition of deletion-reduction depth, deletions were only performed on vertices of degree at least 3 (because others were removed by reduction), here vertices of any degree may be deleted.

4.2 Definition of deletion tree

Just as in the case of deletion-reduction depth, we can construct a rooted tree T with k levels, which we will call a deletion tree of G. The deletion tree is uniquely determined by the sets $V_0, \ldots, V_{k-1}$.

First note that there is a one-to-one correspondence between the vertices of G and the components of all the graphs G_i; we denote by $v(C)$ the vertex deleted from component C and by $C(v)$ the component from which v is deleted.

The tree T is constructed as follows: the vertices of T are the same as the vertices of G. The root of the tree is $v(G_0)$. A vertex v is the parent of each vertex $v(C)$, where C is a component of $C(v) - v$. (Note that the fact that two vertices are adjacent in T does not imply that they are adjacent in G.)

This is essentially the same as a deletion-reduction tree. If d_i is a sequence of deletions (one vertex at a time) of the vertices in V_i, then the concatenation of $d_0, \ldots, d_{k-1}$ is a deletion-reduction sequence containing no reductions. Then the deletion-reduction tree for this sequence (minus the vertex labels, which are all D in any case) is exactly the deletion tree T.

4.3 Deletion depth vs. deletion-reduction depth

We consider the relationship between the values of the two parameters.

Valid reduction sequences

We will use the results of Kneis *et al.* [8]. For a subset D of V, they define a valid reduction sequence to be a sequence of vertices, each of which is either not in D and can be removed by a series-parallel reduction, or is in D and is deleted. They prove that any two maximal valid reduction sequences produce the same graph. We give details below.

We first quote the definition of a reduction from [8] (note that the notation and terminology are slightly different from ours):

Definition of reduction [8]

Let $G = (V, E)$ be a graph and $D \subseteq V$ an arbitrary subset of its nodes. We define the following reduction rules:

R_0: If there is a $v \notin D$ with $\deg(v) = 0$, then remove v.

R_1: If there is a $v \notin D$ with $\deg(v) = 1$, then remove v.

R_2: If there is a $v \notin D$ with $\deg(v) = 2$, then contract v, i.e., remove v and insert a new edge between its two neighbors, if no such edge exists.

R_D: If G contains a node $v \in D$, then remove v.

R: If any of the above rules can be applied, do so.

R^*: Iterate R as long as possible.

Note that it is not obvious that R^* is well-defined, but Kneis *et al.* show that this is so in Theorem 4.1 below.

Now we also quote the definition of a valid reduction sequence.

Definition of valid reduction sequence [8]

Let $G = (V, E)$ be a graph, let $D \subseteq V$, and let $v \in V$ be a node that can be reduced according to R_0, R_1, R_2, or R_D. Then v is called reducible and $G\langle v\rangle$ denotes the graph obtained from G by applying the respective rule on v. For $r \geq 2$ we define $G\langle v_1, \ldots, v_r\rangle = G\langle v_1\rangle\langle v_2, \ldots, v_r\rangle$ inductively. If v_i is reducible in $G\langle v_1, \ldots, v_{i-1}\rangle$ for all $1 \leq i \leq r$, then $(v_1, \ldots, v_r)$ is a valid reduction sequence for G with respect to D. By ε we denote the (valid) empty reduction sequence.

We can now quote a key theorem from [8], which says that any two maximal reduction sequences give the same graph:

Theorem 4.1 (Kneis *et al.* [8]) *Let $G = (V, E)$ be a graph and $D \subseteq V$. Then $R^*(G)$ is well defined; i.e., if τ_1 and τ_2 are two valid reduction sequences for G of maximal length, then $G\langle\tau_1\rangle = G\langle\tau_2\rangle$.* □

As remarked in Section 2.2, it follows by taking $D = \emptyset$ that the reduced graph $r(G)$ is well-defined.

4.4 Upper bounds for deletion depth

We first show that deletion depth is not much larger than deletion-reduction depth.

We will need the fact that series-parallel graphs have small separators. This follows from the fact that series-parallel graphs have treewidth at most two [1], and the following theorem of Robertson and Seymour:

Theorem 4.2 ([12]) *Suppose we have a tree decomposition of G. If the tree decomposition has width $< w$ and $Q \subseteq V(G)$, then there exists $X \subseteq V(G)$ with $|X| \leq w$ such that every component of $G - X$ has at most $\frac{1}{2}|Q - X|$ vertices which are in Q.* □

Lemma 4.3 *Let G be a graph with n vertices. Then* $\mathrm{dd}(G) \leq \mathrm{drd}(G) + O(\log n)$.

Proof. Suppose that $\mathrm{drd}(G) = k$, and let $G_0 = r(G), G_1, \ldots, G_k$, with G_k the empty graph, and vertex sets $V_0, \ldots, V_{k-1}$, satisfy (a) V_{i-1} is a subset of $V(G_{i-1})$ containing at most one vertex from each component of G_{i-1}; (b) $G_i = r(G_{i-1} - V_{i-1})$.

Define the graphs G'_i, $i = 0, \ldots, k$, by $G'_0 = G$ and $G'_i = G'_{i-1} - V_{i-1}$. We will prove that (a) V_i contains at most one vertex from each component of G'_i, and (b) $r(G'_k)$ is empty.

First, it is clear that for each i, $V(G_i) \subseteq V(G'_i)$, hence $V_i \subseteq V(G'_i)$ for each i.

Let $D = V_0 \cup \cdots \cup V_{i-1}$ for some i. Enumerate the vertices in D as $w_1, \ldots, w_t$, where we list first those in V_0, then those in V_1 etc. Then we can define two maximal valid reduction sequences starting from G as follows:

(i) First do series-parallel reduction to form $r(G)$, then alternately delete the next w_j in the enumeration of D, and do series-parallel reduction as long as possible.

Note that none of the series-parallel reductions can be of a vertex in D, since these vertices all have degree at least 3 when deleted (and therefore also earlier in the sequence). Thus this reduction sequence is valid, and clearly maximal.

(ii) Delete in turn all of the vertices in D, then perform series-parallel reductions as long as possible. This sequence is clearly valid and maximal.

By Theorem 4.1, these two sequences lead to the same graph, which is G_i (since sequence (i) corresponds with the definition of $G_0, \ldots, G_i$).

Now deleting $V_0, \ldots, V_{i-1}$ from G results in G'_i, hence sequence (ii) results in the graph $r(G'_i)$. Thus we obtain $r(G'_i) = G_i$. Also V_i contains at most one vertex from each component of G_i. If two vertices are in the same component of G'_i then (if still present) they are also in the same component of $r(G'_i)$. Hence V_i contains at most one vertex from each component of G'_i, as required.

Also, since $r(G'_k) = G_k$ which is empty, we know that G'_k is series-parallel. It follows from Theorem 4.2 (with Q the whole vertex set) that any series-parallel graph has a set of at most 3 vertices whose deletion from the graph leaves all components of size at most $\frac{1}{2}(n-2)$. From this it is easy to show that $\mathrm{dd}(G) \leq 3(\log_2 n) + 1$ for any series-parallel graph G.

It is clear that $\mathrm{dd}(G) \leq k + \mathrm{dd}(G'_k)$. Hence since $k = \mathrm{drd}(G)$ we have $\mathrm{dd}(G) \leq \mathrm{drd}(G) + O(\log n)$, as required. □

In view of Lemma 4.3, the upper bounds on deletion-reduction depth from Section 3 carry over unchanged to deletion depth (the $O(\log n)$ term can be absorbed into existing terms).

Theorem 4.4 *Let G be a graph with m edges, and $r(G)$ be the reduced graph of G. Let n_d be the number of vertices of degree d in $r(G)$. Then*

$$\mathrm{dd}(G) \leq \sum_{v \in V(r(G))} g_{1/5}(d_{r(G)}(v)) + o(|V(G)|) = \sum_{d \geq 3} g_{1/5}(d) n_d + o(|V(G)|)$$

and

$$\mathrm{dd}(G) \leq \frac{9}{50} m + o(m).$$

□

Theorem 4.5 *Let G be a graph of average degree $d \geq 2$, with n vertices and m edges. Then if G is connected, or $d \geq 5$,*

$$\mathrm{dd}(G) \leq (g_{1/5}(d) + o(1))n = (2g_{1/5}(d)/d + o(1))m = (1 - \tfrac{17/5}{d+1} + O(1/d^3))n.$$

□

Somewhat surprisingly, these results show that series-parallel reductions of Max 2-CSP instances are not really needed to obtain the time bounds in Theorems 3.9–3.11. Although series-parallel reductions are used in the construction of the deletion tree (specifically to determine the sets $V_0, \ldots, V_{k-1}$), once the deletion tree for G has been constructed, any instance of Max 2-CSP on G can be solved using only delete and branch operations, treating all vertices, regardless of degree, in the same way.

4.5 $\mathrm{dd}(G) > \mathrm{drd}(G)$

It seems, intuitively, very likely that the deletion depth cannot be smaller than the deletion-reduction depth, and this is indeed the case. To prove this we first need the following lemma.

Lemma 4.6 *Let w be a vertex of G, and H be a subgraph of G. Then one of the following holds (i) $r(H)$ is empty, (ii) $r(H) = r(H')$ for some subgraph H' of $G - w$, or (iii) there exists a vertex v of $r(H)$ such that $r(r(H) - v) = r(H')$ for some subgraph H' of $G - w$.*

Proof. The result is immediate if $r(H)$ is empty, so suppose $r(H)$ is nonempty. If v is a vertex of $r(H)$, then by Theorem 4.1 with $D = \{v\}$, it is easy to see that $r(r(H) - v) = r(H - v)$.

If w is not a vertex of H, then H is a subgraph of $G - w$. Hence (ii) is true with $H' = H$.

If w is a vertex of $r(H)$, then (iii) is true if we choose $v = w$, since $H - w$ is a subgraph of $G - w$.

The remaining case is when w is a vertex of H but not of $r(H)$, i.e. w is removed by the series-parallel reduction of H. So consider a sequence $r_1, \ldots, r_t$ of series-parallel reductions which reduces H to $r(H)$. One of these must remove the vertex w. If this reduction is of type R0, R1 or R2a, then the reduction is identical with the deletion of w. Then taking $D = \{w\}$, this sequence of reductions must give the same result as deleting w and then using series-parallel reductions, so that $r(H) = r(H - w)$. Hence (ii) is true with $H' = H - w$.

So suppose that the reduction r_j which removes w is of type R2n. Since a reduction of type R2n does not change the degree of any other vertex, it is clear that we can postpone this reduction and continue with the other reductions until either a neighbour of w is deleted by a reduction r_k of type R1 or R2a, or no further reductions can be done.

In the first case, the removal of the neighbour of w leaves w with degree 1, so it can then be deleted by a reduction r'_j of type R1. The resulting graph, obtained from the sequence $r_1, \ldots, r_{j-1}, r_{j+1}, \ldots, r_k, r'_j$ is identical with the one obtained from the original sequence $r_1, \ldots, r_k$. Thus the graph $r(H)$, which results from $r_1, \ldots, r_t$, also results from $r_1, \ldots, r_{j-1}, r_{j+1}, \ldots, r_k, r'_j, r_{k+1}, \ldots, r_t$, where r'_j, which removes w, is of type R1. Then from above $r(H) = r(H - w)$ and (ii) is true with $H' = H - w$.

So assume that no further reductions can be done. Then the sequence $r_1, \ldots, r_{j-1}, r_{j+1}, \ldots, r_t$ results in the graph $r(H)$ except for the vertex w subdividing one edge uv say. Then deleting v leaves w with degree 1, so w can then be removed by a reduction of type R1 (which is just a deletion). Further series-parallel reduction gives the graph $r(r(H) - v)$. Let $D = \{v, w\}$ in Theorem 4.1. Then we have a maximal valid reduction sequence which obtains $r(r(H) - v)$ from H. However another maximal valid reduction sequence is to delete w, then delete v, then do series-parallel reductions as long as possible. This clearly gives $r(H - \{v, w\})$, hence by Theorem 4.1, $r(r(H) - v) = r(H - \{v, w\})$. But then (iii) is true with $H' = H - \{v, w\}$, as required. □

DC operations

Define a DC operation to be the deletion of at most one vertex from each component of a graph (but at least one vertex overall).

Lemma 4.7 *Let G be a nonempty graph, and H a subgraph of G. Let $\phi_1, \ldots, \phi_k$ be a sequence of operations each of which is one of R0, R1, R2n, R2a, or DC. Suppose that $G^{(0)} = G$, $G^{(i)} = \phi_i(G^{(i-1)})$ for $i = 1, \ldots, k$, and $G^{(k)}$ is empty. Then* $\mathrm{drd}(H) \leq t$, *where t is the number of DC operations in the sequence.*

Proof. We will use induction on k. If $k = 0$, then G is the empty graph, and then H is empty , so $\mathrm{drd}(H) = 0$ as required. So suppose that $k > 0$, and that G has s components $C_1, \ldots, C_s$. Let the corresponding components of H be $H_1, \ldots, H_s$ (some of these may be empty).

First suppose that ϕ_1 is one of R0, R1, R2n, R2a, and let v be the vertex removed by ϕ_1. If H does not contain v, then H is a subgraph of $G^{(1)}$. If H does contain v, then it is easy to see that we can choose an operation ϕ, one of R0, R1, R2n, R2a, so that $\phi(H)$ is a subgraph of $G^{(1)}$. Since $G^{(1)}$ can be transformed into the empty graph by a sequence of $k-1$ operations, then by the inductive hypothesis we have in the first case $\mathrm{drd}(H) \leq t$, as required. In the second case we have $\mathrm{drd}(\phi(H)) \leq t$, and since $\mathrm{drd}(\phi(H)) = \mathrm{drd}(r(H)) = \mathrm{drd}(H)$, the result follows.

If ϕ_1 is a DC operation, there is a set V_0 containing at most one vertex from each component C_j, such that $G_0 - V_0$ can be transformed into the empty graph by a sequence of $k-1$ operations, of which $t-1$ are DC operations. Suppose that V_0 contains a vertex w_j in component C_j. Then from Lemma 4.6, one of the following holds (i) $r(H_j)$ is empty, (ii) $r(H_j) = r(H'_j)$ for some subgraph H'_j of $G_j - w_j$, or (iii) there exists a vertex v_j of $r(H_j)$ such that $r(r(H_j) - v_j) = r(H'_j)$ for some subgraph H'_j of $G_j - w_j$. Let V'_0 consist of all the vertices v_j, for those components C_j for which (iii) holds. Then $r(r(H) - V'_0)$ is equal to $r(H')$ for some subgraph H' of $G_0 - V_0$. By the inductive hypothesis, $\mathrm{drd}(H') \leq t-1$, hence $\mathrm{drd}(r(H')) \leq t-1$. But then $r(r(H) - V'_0)$ has deletion-reduction depth at most $t-1$, and V'_0 contains at most one vertex from each component of $r(H)$, so by the definition of deletion-reduction depth, we have $\mathrm{drd}(H) \leq t$, as required. □

Remarks: (1) In the definition of deletion-reduction depth, we required that the graph be fully reduced between successive DC operations. The lemma above shows that no advantage would be gained by allowing arbitrary sequences of series-parallel reductions between successive DC operations. (2) Similarly, it follows that $\mathrm{drd}(G)$ equals the minimum, over all possible deletion-reduction trees for G, of the maximum number of deletions on a path from the root to a leaf (previously we had only established that $\mathrm{drd}(G)$ was an upper bound for this quantity).

Lemma 4.8 *Let G be a nonempty graph, and H a subgraph of G. Then* $\mathrm{drd}(H) < \mathrm{dd}(G)$.

Proof. By definition, G can be transformed into the empty graph by a sequence of $\mathrm{dd}(G)$ DC operations. Hence before the last of these operations, the graph must have no edges, so the last DC operation can be replaced by a sequence of R0 operations. Then by Lemma 4.7, $\mathrm{drd}(H) \leq \mathrm{dd}(G) - 1$, as required. □

Corollary 4.9 *Let G be a nonempty graph. Then* $\mathrm{drd}(G) < \mathrm{dd}(G)$.

Proof. Take $H = G$ in Lemma 4.8. □

Lemma 4.10 *Let G be a nonempty graph, and H a subgraph of G. Then* $\mathrm{drd}(H) < \mathrm{drd}(G)$.

Proof. By definition, G can be transformed into the empty graph by a sequence consisting of reductions (R0, R1, R2a, R2n) and at most $\mathrm{drd}(G)$ DC operations. Then by Lemma 4.7, $\mathrm{drd}(H) \leq \mathrm{drd}(G)$, as required. □

Lemma 4.11 *Let G be a graph, and X a subset of the vertices of G. Then* $\mathrm{drd}(G) \leq |X| + \mathrm{drd}(G - X)$.

Proof. Consider the sequence of operations consisting of (i) $|X|$ DC operations, deleting one element of X at a time, followed by (ii) a sequence of deletions and reductions containing at most $\mathrm{drd}(G - X)$ DC operations and resulting in the empty graph (this exists by the definition of $\mathrm{drd}(G - X)$). This contains a total of $|X| + \mathrm{drd}(G - X)$ DC operations, so the result follows from Lemma 4.7. □

4.6 A lower bound for deletion depth of cubic graphs

Kostochka and Melnikov [9] show that almost every cubic graph G has bisection width $\mathrm{bw}(G) \geq \frac{1}{9.9}n$. We can use this to obtain a linear lower bound on the deletion depth of cubic graphs.

We first bound the bisection width in terms of the deletion depth and maximum degree.

Lemma 4.12 *Let G be a connected graph with maximum degree Δ. Then* $\mathrm{bw}(G) \leq \lfloor \frac{\Delta}{2} \rfloor (\mathrm{dd}(G) - 1) + \lfloor \frac{\Delta+1}{2} \rfloor (\Delta - 2)$.

Proof. Let $k = \mathrm{dd}(G)$. Then there is a sequence $G_0 = G, G_1, \ldots, G_k$, with G_k the empty graph, and a sequence $V_0, \ldots, V_{k-1}$ of vertex sets, such that for each $i = 1, \ldots, k$, V_{i-1} is a subset of $V(G_{i-1})$ containing one vertex from each component of G_{i-1}, and $G_i = G_{i-1} - V_{i-1}$. Let T be the corresponding deletion tree for G.

We impose a linear ordering on the vertices of T (and G) as follows: for each vertex v of T, the children of v are the roots of subtrees. We order these subtrees arbitrarily and make them all greater than v in the ordering (this is the usual pre-order). Enumerate the vertices as $v_1 \leq v_2 \leq \cdots \leq v_n$.

Then for each vertex v_i define a partition π_i of $V(G)$ by setting $A_i = \{v \mid v \leq v_i\}$ and $B_i = \{v \mid v > v_i\}$. For any node u of T, let $P(u)$ be the path in T from u to the root.

Now consider an edge vw in G with $v \in A_i$ and $w \in B_i$, so that $v \leq v_i < w$. Let y be the node in T where the paths $P(v)$ and $P(w)$ meet. If $y \neq v, w$, then v and w are in different components of $G - V(P(y))$, which is impossible since vw is an edge. We cannot have $y = w$, for then v is a descendent of w, contradicting $v < w$. Hence $y = v$ and w is a descendent of v. Let z be the node where the paths $P(v)$ and $P(v_i)$ meet, and suppose that $z \neq v$. Then we cannot have $z = v_i$, for then v is a strict descendent of v_i, contradicting $v \leq v_i$. So suppose that $z \neq v_i$. Then v, v_i are in subtrees T_1, T_2 respectively of z, and since $v \leq v_i$, T_1 comes before T_2 in the ordering. However since w is a descendent of v, this means that w is in T_1 also, so $w < v_i$, a contradiction. So we must have $z = v$, hence v is on the path $P(v_i)$ in T.

Thus for every edge vw in G with $v \in A_i$ and $w \in B_i$, the endpoint in A_i, i.e., v, is on the path $P(v_i)$. Also if v_i is a leaf of the tree, then v cannot be equal to v_i (because w is a descendent of v). Hence since $|V(P(v_i))| \leq \mathrm{dd}(G)$, at most $\mathrm{dd}(G) - 1$ vertices of A_i are adjacent in G to a vertex of B_i.

Now for each $j = 1, \ldots, \Delta$, let $X_i^{(j)}$ be the set of vertices of A_i which have j neighbours in G in the set B_i. Thus $\sum_{j=1}^{\Delta} |X_i^{(j)}| \leq \mathrm{dd}(G) - 1$. Also $\mathrm{cut}(\pi_i) = \sum_{j=1}^{\Delta} j|X_i^{(j)}|$.

Now let $Z_i = \bigcup_{j=\lfloor \Delta/2 \rfloor + 1}^{\Delta} X_i^{(j)}$. Thus Z_i consists of those vertices of A_i which have more neighbours in B_i than in A_i. For any two disjoint sets U, W of vertices, let $E(U, W)$ be the set of edges with one endpoint in U and the other in W. Then from the definition of $X_i^{(j)}$, we have

$$|E(Z_i, B_i)| = \sum_{j=\lfloor \Delta/2 \rfloor + 1}^{\Delta} j|X_i^{(j)}|,$$

hence

$$|E(Z_i, A_i - Z_i)| \leq |E(Z_i, A_i)| \leq \sum_{j=\lfloor \Delta/2 \rfloor + 1}^{\Delta} (\Delta - j)|X_i^{(j)}|.$$

Now form a new partition $\pi_i' = (A_i', B_i')$ by moving the vertices in Z_i from A_i to B_i, i.e. set $A_i' = A_i - Z_i$ and $B_i' = B_i \cup Z_i$. Then

$$|E(A_i', B_i')| = |E(A_i - Z_i, B_i)| + |E(A_i - Z_i, Z_i)|.$$

Hence we have

$$\begin{aligned}
\mathrm{cut}(\pi_i') &\leq \sum_{j=1}^{\lfloor \Delta/2 \rfloor} j|X_i^{(j)}| + \sum_{j=\lfloor \Delta/2 \rfloor + 1}^{\Delta} (\Delta - j)|X_i^{(j)}| \\
&\leq \lfloor \Delta/2 \rfloor \sum_{j=1}^{\Delta} |X_i^{(j)}| \\
&\leq \lfloor \Delta/2 \rfloor \, (\mathrm{dd}(G) - 1).
\end{aligned}$$

Thus each of these cuts is of approximately the right size, but we also need the partition to be balanced. To achieve this we first choose I so that partition π_I' is as nearly balanced as possible. We have $|A_i'| = i - \sum_{j=\lfloor \Delta/2 \rfloor + 1}^{\Delta} |X_i^{(j)}|$, so that $|A_i'| = \lfloor n/2 \rfloor + r_i$, where $r_i = i - \lfloor n/2 \rfloor - \sum_{j=\lfloor \Delta/2 \rfloor + 1}^{\Delta} |X_i^{(j)}|$. Thus we want to choose I such that r_I is as small as possible.

First note that $r_{i+1} - r_i \leq \Delta + 1$, since at most Δ vertices in $\bigcup_{j=\lfloor \Delta/2 \rfloor + 1}^{\Delta} X_i^{(j)}$ may be neighbours of v_{i+1} and so absent from $\bigcup_{j=\lfloor \Delta/2 \rfloor + 1}^{\Delta} X_{i+1}^{(j)}$.

Also $r_{\lfloor n/2 \rfloor} \leq 0$, while each set $X_n^{(j)}$ is empty (since B_n is empty) so that $r_n = \lceil n/2 \rceil > 0$. Then it follows easily that there is an integer $I \geq \lfloor n/2 \rfloor$ such that $|r_I| \leq \lfloor \frac{\Delta+1}{2} \rfloor$.

Thus we have $|A_I'| = \lfloor n/2 \rfloor + r_I$, where $|r_I| \leq \lfloor \frac{\Delta+1}{2} \rfloor$. Then we can obtain a balanced partition of G by moving at most $\lfloor \frac{\Delta+1}{2} \rfloor$ vertices from A_I' to B_I' or from B_I' to A_I'. Provided we choose these to be endpoints of edges in the cut, this will increase the size of the cut

π'_I by at most $\lfloor \frac{\Delta+1}{2} \rfloor (\Delta - 2)$, hence we obtain

$$\mathrm{bw}(G) \leq \left\lfloor \frac{\Delta}{2} \right\rfloor (\mathrm{dd}(G) - 1) + \left\lfloor \frac{\Delta + 1}{2} \right\rfloor (\Delta - 2).$$

□

Corollary 4.13 *Let G be a connected cubic graph. Then* $\mathrm{bw}(G) \leq \mathrm{dd}(G) + 1$. □

Note that the cubic graph $K_{3,3}$ has deletion depth 4 and bisection width 5, so this bound is best possible.

Applying the result of Kostochka and Melnikov [9] that almost every cubic graph G has $\mathrm{bw}(G) \geq \frac{1}{9.9}n$, we obtain the following.

Corollary 4.14 *Almost all cubic graphs G satisfy* $\mathrm{dd}(G) \geq \frac{1}{9.9}|V(G)| - 1$. □

Since $\mathrm{drd}(G) \geq \mathrm{dd}(G) - O(\log n)$ by Lemma 4.3, we have

Corollary 4.15 *Almost all cubic graphs G satisfy* $\mathrm{drd}(G) \geq \frac{1}{9.95}|V(G)|$. □

An improvement on the upper bound of $\frac{1}{5}n + o(n)$ from Lemma 3.4 would lead to a faster algorithm for Max 2-CSP. Thus it would be of interest to bring the upper and lower bounds closer together.

Appendix

We now prove the properties of the function $\overline{\max}$ defined in Lemma 3.3. Recall that for any real numbers x, y,

$$\overline{\max}(x, y) = \min(\max(x, y - 1), \max(x - 1, y)).$$

First note the following:

A. $\overline{\max}(x + a, y + a) = \overline{\max}(x, y) + a$.
This follows from the corresponding property of max and min.

B. $\overline{\max}(x, y) = \max(\max(x, y) - 1, \min(x, y))$.
Set $M = \max(x, y)$ and $m = \min(x, y)$. Then clearly $\overline{\max}(x, y) = \overline{\max}(M, m)$, and

$$\begin{aligned} \overline{\max}(M, m) &= \min(\max(M, m - 1), \max(M - 1, m)) \\ &= \min(M, \max(M - 1, m)) \\ &= \max(M - 1, m). \end{aligned}$$

Then the properties of $\overline{\max}$ can be verified as follows:

M1. $\overline{\max}$ is monotone non-decreasing in each argument.
This follows from the monotonicity of max and min.

M2. $\max(x, y) \geq \overline{\max}(x, y) \geq \max(x, y) - 1$.
This follows immediately from B.

M3. If $|x - y| \leq 1$, $\overline{\max}(x, y) = \min(x, y)$.
This also follows immediately from B.

M4. $\overline{\max}(x - 1, y + 1) \leq \overline{\max}(x, y) + 1$.

$$\begin{aligned} \overline{\max}(x, y) + 1 &= \overline{\max}(x + 1, y + 1) && \text{(by A)} \\ &\geq \overline{\max}(x - 1, y + 1) && \text{(by M1).} \end{aligned}$$

M5. If $x' \leq \max(x, y) - 3$, and $y' \leq \min(x, y) - 1$, then $\overline{\max}(x', y') \leq \overline{\max}(x, y) - 2$.

$$\begin{aligned} \overline{\max}(x', y') &\leq \max(x', y' - 1) && \text{(by definition of } \overline{\max}) \\ &= \max(x' + 2, y' + 1) - 2 \\ &\leq \max(\max(x, y) - 1, \min(x, y)) - 2 \\ &= \overline{\max}(x, y) - 2 && \text{(by B).} \end{aligned}$$

Acknowledgements

I would like to thank the anonymous referee for a very careful reading of the paper and a number of helpful comments.

References

[1] H. L. Bodlaender, A partial k-arboretum of graphs with bounded treewidth, *Theoretical Computer Science* **209** (1998) 1–45.

[2] K. J. Edwards and G. E. Farr, Improved upper bounds for planarization and series-parallelization of average degree bounded graphs, *Electronic Journal of Combinatorics* **19 (2)** (2012) #P25.

[3] K. J. Edwards and E. McDermid, A general reduction theorem with applications to pathwidth and the complexity of MAX 2-CSP, *Algorithmica*, to appear.

[4] F. V. Fomin and K. Høie, Pathwidth of cubic graphs and exact algorithms, *Inf. Process. Lett.* **97** (2006) 191–196.

[5] S. Gaspers and G. B. Sorkin, A universally fastest algorithm for Max 2-Sat, Max 2-CSP, and everything in between, *J. Comput. System Sci.* **78** (2012), 305–335.

[6] S. Gaspers and G. B. Sorkin, Separate, Measure and Conquer: Faster Algorithms for Max 2-CSP and Counting Dominating Sets, (2014), arXiv:1404.0753v1 [cs.DS].

[7] A. Golovnev and K. Kutzkov, New Exact Algorithms for the 2-Constraint Satisfaction Problem, unpublished manuscript.
`http://cims.nyu.edu/~golovnev/papers/max2csp.pdf`

[8] J. Kneis, D. Mölle, S. Richter and P. Rossmanith, A bound on the pathwidth of sparse graphs with applications to exact algorithms, *SIAM Journal on Discrete Mathematics* **23** (2009) 407–427.

[9] A.V. Kostochka, L.S. Melnikov, On a lower bound for the isoperimetric number of cubic graphs, in: V.F. Kolchin et al. (Eds.), Probabilistic Methods in Discrete Mathematics, Proc. third Int. Petrozavodsk Conf., Progress in Pure and Applied Discrete Mathematics, vol. 1, TVP/VSP, 1993, pp. 251–265.

[10] A. S. Kulikov and S. S. Fedin, Solution of the maximum cut problem in time $2^{|E|/4}$ (Russian), Zap. Nauchn. Sem. S.-Peterburg. Otdel. Mat. Inst. Steklov. (POMI) **293** (2002), Teor. Slozhn. Vychisl. 7, 129–138, 183; translation in J. Math. Sci. (N. Y.) **126** (2005), 1200–1204.

[11] B. Monien and R. Preis, Upper bounds on the bisection width of 3- and 4-regular graphs, *J. Discrete Algorithms* **4** (2006) 475–498.

[12] N. Robertson and P.D. Seymour, Graph minors. II. Algorithmic aspects of tree-width, *J. Algorithms* **7** (1986) 309–322.

[13] A. D. Scott and G. B. Sorkin, Faster algorithms for MAX CUT and MAX CSP, with polynomial expected time for sparse instances, in: *Proc. 7th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM 2003)*, Lecture Notes in Computer Science **2764**, Springer, 2003, pp. 382–395.

[14] A. D. Scott and G. B. Sorkin, A faster exponential-time algorithm for Max 2-Sat, Max Cut, and Max k-Cut, Tech. Report RC23457 (W0412-001), IBM Research Report, December 2004.

[15] A. D. Scott and G. B. Sorkin, Linear-programming design and analysis of fast algorithms for Max 2-CSP, *Discrete Optimization* **4** (2007) 260–287.

[16] A. D. Scott and G. B. Sorkin. Polynomial constraint satisfaction problems, graph bisection, and the Ising partition function, *ACM Trans. Algorithms* **5** (2009) Art. 45.

[17] R. Williams, A new algorithm for optimal constraint satisfaction and its implications, in: J. Díaz *et al.* (eds.), *Proc. 31st International Colloquium on Automata, Languages and Programming (ICALP)* (Turku, Finland, 2004), Lecture Notes in Computer Science **3142**, Springer, 2004, pp. 1227–1237.