Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 101 (2015)

On the structure of Solution-Graphs for Boolean Formulas*

Patrick Scharpfenecker†

University of Ulm, Institute of Theoretical Computer Science,
Ulm, Germany
patrick.scharpfenecker@uni-ulm.de

August 10, 2015

Abstract

In this work we extend the study of solution graphs and prove that for boolean formulas in a class called CPSS, all connected components are partial cubes of small dimension, a statement which was proved only for some cases in [17]. In contrast, we show that general Schaefer formulas are powerful enough to encode graphs of exponential isometric dimension and graphs which are not even partial cubes.

Our techniques shed light on the detailed structure of *st*-connectivity for Schaefer and connectivity for CPSS formulas, problems which were already known to be solvable in polynomial time. We refine this classification and show that the problems in these cases are equivalent to the satisfiability problem of related formulas by giving mutual reductions between (*st*-)connectivity and satisfiability. An immediate consequence is that *st*-connectivity in (undirected) solution graphs of Horn-formulas is P-complete while for $2SAT$ formulas *st*-connectivity is NL-complete.

1 Introduction

The work of Schaefer [15] first introduced a dichotomy for the complexity of satisfiability on different classes of boolean formulas. The author proved that for specific boolean formulas (now called Schaefer formulas), satisfiability is in P while for all other classes, satisfiability is NP-complete. Surprisingly, there are no formulas of intermediate complexity. Recently, the work of Gopalan et al. and Schwerdtfeger [8, 17] uncovered a similar behavior for several problems on solution graphs of boolean formulas. A solution graph is a subgraph of the n-dimensional hypercube induced by all satisfying assignments, see Definition 1. Therefore boolean formulas can be seen as a succinct encoding of a solution graph.

*A full version of this paper [16] will appear in Proceedings of the 20th International Symposium on Fundamentals of Computation Theory, 2015.

†Supported by DFG grant TO 200/3-1.

ISSN 1433-8092

Definition 1. *Let $F(x_1, \ldots, x_n)$ be an arbitrary boolean formula. Then the solution graph G_F is the subgraph of the n-dimensional hypercube H_n induced by all satisfying solutions x of F.*

These works focused on classifying the complexity of the connectivity and *st*-connectivity problem on solution graphs for given classes of formulas. While *st*-connectivity is the problem to determine for a given graph and two nodes if there is a path between these nodes, connectivity asks if a given graph consists only of a single connected component.

Usually, succinct encodings provide a complexity blow-up compared to non-succinct encodings (see for example [14, 3, 18, 19, 5]). Therefore the question arises to what extent the complexity for *st*-connectivity and connectivity change in the case of solution graphs in relation to the power of the encoding formulas.

For this, Gopalan et al. [8] introduced a new class of boolean formulas they call "tight" which lies between Schaefer formulas and general formulas. Their classification shows that for tight formulas, *st*-connectivity is in P while for general formulas it is PSPACE-complete. Similar, for the connectivity problem, they achieve a coNP-algorithm for Schaefer formulas, coNP-completeness for tight formulas and PSPACE-completeness for general formulas and conjecture that this is actually a trichotomy: they suspected connectivity for Schaefer formulas to be in P. More recently, [17] proved a trichotomy by introducing a fourth class of formulas (besides Schaefer, tight and general formulas) the authors call CPSS (constraint-projection separating Schaefer) which is even more restrictive than Schaefer formulas and by modifying the definition of tight formulas to "safely tight" formulas. Figure 1 summarizes the results of [17] and [8].

Function Set R	***Conn(R)***	***stConn(R)***	Diameter
CPSS	P	P	$O(n)$
Schaefer, not CPSS	coNP-complete	P	$O(n)$
safely tight, not Schaefer	coNP-complete	P	$O(n)$
not safely tight	PSPACE-complete	PSPACE-complete	$2^{\Omega(\sqrt{n})}$

Figure 1: Classification of Connectivity Problems.

We refine the P-algorithms for *st*-connectivity of tight formulas (which contains all safely tight formulas) and show a close relation to satisfiability of such formulas by an improved analysis of the structure of solution graphs. For all tight formulas, *st*-connectivity reduces to satisfiability of a related formula. So for example, *st*-connectivity on $2SAT$ and $Horn$ formulas can be reduced to satisfiability of the same type. Therefore in the first case, *st*-connectivity is in NL while for the second case, the P-algorithm seems tight. In addition, for $2SAT$ and Horn-formulas, the reverse holds too, that is, satisfiability for these formulas is reducible to *st*-connectivity and connectivity in the solution graph of the same type of formulas. So *stConn(2SAT)* is NL-complete and *stConn($Horn_3$)* is P-complete.

While [8] proved that for all tight formulas the diameter of connected components is linearly bounded in the number of variables and [17] improved this by showing that bijunctive formulas are even partial cubes, there is still room for improvements. Thereby a partial cube is an induced graph of the hypercube which preserves distances. So if two nodes are connected, their distance in the partial cube has to be the same as in the original hypercube. In our work we study the structure of connected components in solution graphs of Schaefer formulas. For CPSS formulas we show that every connected component is a partial cube of small dimension[1] while general Schaefer formulas are powerful enough to encode partial cubes of exponential dimension or even graphs which are not partial cubes at all. Yet these graphs have still diameter bounded by $O(n)$.

We note that the work of Ekin [6] discusses similar properties like connectedness and geodesy based on the structure of a given DNF formula. The authors discuss recognition of these properties and give a hierarchy of boolean functions which admit these properties. While co-geodetic functions are connected partial cubes, their approach requires the input formula to be a DNF or CNF. In contrast, the work of [8, 17] can use arbitrary boolean formulas as clauses.

Another related topic is the so called phase-transition for random kSAT formulas and the clustering of the solution space. The works of [11, 12, 10] shed light on the behaviour of random formulas by providing a threshold α_c implying that random kSAT with less than $\alpha_c \cdot n$ clauses on n variables are most likely satisfiable while more than $\alpha_c \cdot n$ clauses imply that the formula is most likely unsatisfiable. Further, the authors of [12] showed that there is another threshold $\alpha_d \leq \alpha_c$ such that formulas with density lower than α_d mainly encode single connected components while formulas with density between α_d and α_c encode many connected components, called clusters. The work of [8, 17] and our work can be seen as stepping stones to a better understanding of the structure of solution graphs which may help analyzing the structure of solution graphs of random kSAT formulas.

The rest of this paper is organized as follows. In Section 2 we briefly introduce our notation and basic definitions. Section 3 will cover the characterization of CPSS solution graphs as collections of partial cubes. In contrast, Section 4 will show that general Schaefer-formulas are powerful enough to encode partial cubes of exponential dimension and even graphs which are not partial cubes at all. Finally, in Section 5, we establish reductions from connectivity and st-connectivity problems of solution graphs to the satisfiability problem on related formulas and thereby refine the previous P characterization. We complete the classification of these problems by giving matching lower bounds.

2 Preliminaries

To compare two words $x, y \in \{0,1\}^n$, we use the lexicographic order. For $x, y \in \{0,1\}^n$, $\Delta(x, y)$ denotes the Hamming-distance between x and y and $\Delta(x) := \Delta(x, 0^{|x|})$ denotes the Hamming-weigth of x. We associate words in $\{0,1\}^n$ with subsets of $[n] = \{1, \ldots, n\}$ in the standard way. We use graphs $G = (V, E)$ with nodes $V = [n]$ and edge set $E \subseteq V^2$ without self-loops.

With P we denote the set of decision problems which can be solved in polynomial-time while L (NL) problems can be solved in (non-deterministic) logarithmic-space. With $\leq_T^{\mathrm{L}}$, $\leq_m^{\mathrm{L}}$ and $\leq_m^{\mathrm{P}}$ we denote logarithmic-space Turing and logarithmic-space as well as polynomial-time many-one reductions.

[1] This was proved for bijunctive formulas in [17], we prove the remaining cases of Horn and dual-Horn formulas.

We recall Definition 1 and note that we talk of solution graphs with the Hamming-distance, implying that two satisfying solutions are connected by an edge iff they differ in exactly one variable. Given a graph G and two nodes u, v, $d(u, v)$ is the length of the shortest path between u and v in G or ∞ if there is no such path.

Definition 2. *An induced subgraph G of H_n is a partial cube iff for all $x, y \in G$, $d(x, y) = \Delta(x, y)$. We call such an induced subgraph "isometric".*

For a $2SAT$ formula $F(x_1, \ldots, x_n)$ we define the implication graph $I(F) = (V, E)$ on nodes $V = \{x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}\}$ such that $(k \to l) \in E$ with $k, l \in V$ iff $F \models (k \to l)$.

For all boolean functions $F : \{0,1\}^n \to \{0,1\}$ we can represent F with the subset of all its satisfying assignments in $\{0,1\}^n$. Then a boolean function $F \subseteq \{0,1\}^n$ is closed under a ternary operation $\odot : \{0,1\}^3 \to \{0,1\}$ iff $\forall x, y, z \in F : \odot(x, y, z) := (\odot(x_1, y_1, z_1), \ldots, \odot(x_n, y_n, z_n)) \in F$. Note that we extend the notation of a ternary operation to an operation on three bit-vectors by applying the operation bitwise on the three vectors. We can define a similar closure for binary operations. For R a finite set of boolean functions with arbitrary arities (for example $R = \{(\overline{x} \vee y), (x \oplus y), (x \oplus y \oplus z)\}$, we define $SAT(R)$ to be the satisfiability problem for all boolean formulas which are conjunctions of instantiations of functions in R. For the given example R, $F(x, y, z) = (\overline{z} \vee y) \wedge (x \oplus y)$ is a formula in which every clause is an instantiation of an R-function. With *Conn(R)* we denote the connectivity problem, given a conjunction F of R-functions, is the solution graph connected? Similarly, *stConn(R)* is the *st*-connectivity problem, given a conjunction F of R-functions and s, t, is there a path from s to t in the solution graph? We mostly use F for boolean formulas/functions and R, S for finite sets of functions.

Note that $r \in R$ can be an arbitrary boolean function as for example $r = (x \oplus y)$ or $r = (x \vee \overline{y} \vee \overline{z}) \wedge (\overline{x} \vee z)$. With $2SAT$ we denote the set of all CNF-clauses with two variables and with $Horn_n$ we define the set of all Horn-clauses of size up to n. The ternary majority function $maj : \{0,1\}^3 \to \{0,1\}$ is defined as $maj(a, b, c) = (a \wedge b) \vee (a \wedge c) \vee (b \wedge c)$.

In definitions 3 to 5 we recall some terms which were partially introduced by [17] and [8].

Definition 3. *A boolean function F is*

- *bijunctive, iff it is closed under $maj(a, b, c)$.*
- *affine, iff it is closed under $a \oplus b \oplus c$.*
- *Horn, iff it is closed under $a \wedge b$.*
- *dual-Horn, iff it is closed under $a \vee b$.*
- *IHSB−, iff it is closed under $a \wedge (b \vee c)$.*
- *IHSB+, iff it is closed under $a \vee (b \wedge c)$.*

A function has such a property componentwise, iff every connected component in the solution graph is closed under the corresponding operation. A function F has the additional property "safely", iff the property still holds for every function F' obtained by identification of variables[2].

[2]Identifying two variables corresponds to replacing one of them with the other variable.

In the case of Horn-formulas, the usual definition (the conjunction of Horn-clauses, which is a conjunction of literals such that no two literals occur positive) implies that the represented functions are Horn.

Definition 4. *A set of functions R is Schaefer (CPSS) if at least one of the following conditions holds:*

- *every function in R is bijunctive.*
- *every function in R is Horn (and safely componentwise IHSB−).*
- *every function in R is dual-Horn (and safely componentwise IHSB+).*
- *every function in R is affine.*

If we have a boolean formula F which is build from a set of CPSS functions R we say that F is CPSS. Clearly, every CPSS formula is Schaefer. We later use a bigger class of functions which we call *tight.* This class properly contains all Schaefer sets of functions.

Definition 5. *A set R of functions is tight if at least one of the following conditions holds:*

- *every function in R is componentwise bijunctive.*
- *every function in R is OR-free.*
- *every function in R is NAND-free.*

A function is OR-free if we can not derive $(x \vee y)$ by fixing variables. Similar, a function is NAND-free if we can not derive $(\overline{x} \vee \overline{y})$ by fixing variables.

3 Structure of CPSS-formulas

We now study and refine the properties of connected components in formulas F on n variables which are CPSS. We are going to prove that such connected components are always partial cubes of isometric dimension at most n. Hereby the isometric dimension is the smallest n such that the graph can be isometrically embedded into the hypercube H_n. For this, [8] gives some useful basic properties for bijunctive and affine functions:

Lemma 1 ([8])**.** *If a boolean function F is bijunctive or affine then it is componentwise bijunctive.*

Lemma 2 ([8])**.** *Let F be a componentwise bijunctive function. Then the distance of all solutions x and y in the same connected component is exactly $\Delta(x, y)$.*

Lemma 3 ([8])**.** *Let R be a set of Horn-functions and let F be built from R-functions. Then every connected component in F has a unique minimal solution x^* and every other solution in this component is connected to x^* with a monotone path with respect to the Hamming distance.*

We can now prove our first statement:

Lemma 4. *Given a CPSS formula F, for two satisfying assignments s and t. Either $d(s,t) = \Delta(s,t)$ or $d(s,t) = \infty$.*

Proof. If F is bijunctive or affine the statements follows by Lemma 1 and 2.

Now suppose F is Horn and componentwise IHSB$-$ (the last case is dual). Therefore every connected component in F is closed under $x \wedge (y \vee z)$. We only show that for all x, y with $y \leq x$ in the same component there is a path of length $\Delta(x,y)$. Obviously there can not be a shorter path. With this, the statement holds for all a, b: We just use $c = a \wedge b$ as intermediate step and c is in the same connected component: By Lemma 3, we know that every component in a Horn formula has a unique minimal solution x^*. Then $a \wedge (b \vee x^*) = a \wedge b = c$ is in the same component as a and b.

Suppose by contradiction there is no such path from x to y. Then we know that y can not be the unique minimal solution x^*. But then there is a monotone decreasing path from x to x^* which has to bypass y and decrease at least one variable $i \in x \setminus y$. Let a be the first such node below x which decreases exactly one $i \in x \setminus y$. For all other decreased bits we know that $y_j = 1$. Then $x \wedge (a \vee y) = x \setminus \{i\} = x'$ and $d(x', y) = d(x,y) - 1$. An induction over the distance proofs our statement. □

Corollary 1. *Given a CPSS formula $F(x_1, \ldots, x_n)$, every connected component of F is a partial cube of isometric dimension at most n.*

4 Structure of general Schaefer-Formulas

Previously we looked at properties of solution graphs of Schaefer functions which are in addition CPSS. If a given formula F on n variables is CPSS, every connected component is a partial cube of small isometric dimension. If it is Schaefer but not CPSS, the diameter is still linear in n and due to [8], *st*-connectivity is in P. We now prove that there are Schaefer formulas which encode a partial cube of exponential isometric dimension or even a graph which is not a partial cube at all.

To achieve this, we first create some tools using matrices and their rank. We only use the rank of a matrix with respect to $\mathbb{Z}$. A metric space is a set of elements R equipped with a distance function $d : R \times R \to \mathbb{N}$. We say a matrix $M \in \mathbb{N}^{I \times I}$ with index-set I embeds into a metric space if there is a mapping $\pi : I \to R$ such that for all $i, j \in I$ $M_{i,j} = d(\pi(i), \pi(j))$. An example for such a metric space is the k dimensional integer grid equipped with the L^1-norm (sometimes called Manhattan-norm).

Lemma 5. *The matrix $M = (m_{i,j})_{i,j \in \{0,1\}^n}$ with $m_{i,j} = \Delta(i,j) + 2$ if $i \neq j$ and $m_{i,i} = 0$ has rank at least $2^n - n - 1$.*

Proof. We decompose M as $M = M_1 + M_2$ with $M_1 = (\Delta(i,j))_{i,j \in \{0,1\}^n}$ and $M_2 = M - M_1$. It can be verified that $rank(M_2) = 2^n$ and $rank(M_1) = n + 1$. For the latter, a complete basis consists of all the row-vectors of bit-strings w with $\Delta(w) \leq 1$. We denote these vectors as w^0 for the string of weight 0 and w^i the vector for the string setting bit i to 1. Then a row vector of an arbitrary string a can be computed as $(\sum_{i \in [n]} a_i \cdot w^i) - (\Delta(a) - 1) \cdot w^0$. For a given column b, every $a_i \neq 0$ adds $\Delta(b) - 1$ iff $b_i = 1$ and $\Delta(b) + 1$ otherwise. The sum adds up to $\Delta(a)\Delta(b)+\Delta(a\setminus b)-\Delta(a\cap b)$. As $w^0[b]=\Delta(b)$, we subtract $(\Delta(a)-1)\Delta(b)$. So the result is $\Delta(b)+\Delta(a\setminus b)-\Delta(a\cap b)=\Delta(a\setminus b)+\Delta(b\setminus a)=\Delta(a,b)$.

We know that $rank(-M_1)=rank(M_1)$, and by subadditivity $rank(M_2)\leq rank(M)+rank(-M_1)$. Then $2^n\leq rank(M)+n+1$ and $rank(M)\geq 2^n-n-1$. □

Lemma 6. *If a given point set P with distances $M\in\mathbb{N}^{P^2}$ can be mapped into the metric space $R=\{0,1\}^m$ with L^1 as distance-norm, then $rank(M)\leq 2m$.*

Proof. For a given P we look at the labeling $\pi: P\to R$. Then $d(u,v)=\sum_{i\in[m]}|\pi(u)_i-\pi(v)_i|$. So basically $M=\sum_{i\in[m]}M_i$ with $M_i=(m_i^{j,k})$ and $m_i^{j,k}=|\pi(j)_i-\pi(k)_i|$. So all M_i are $0,1$ block matrices. They define sets $A,B\subseteq P$ such that all entries $(a,b)\in A\times B$ are assigned to 1 and everything else is 0. As M_i is symmetric, we can split M_i into two matrices of rank 1: The first one contains all non-negative entries $(a,b)\in A\times B$ and the second one all $(b,a)\in B\times A$. This implies that M is the sum of $2m$ rank 1 matrices and therefore $rank(M)\leq 2m$. □

Corollary 2. *The matrix $M=(m_{i,j})^{i,j\in\{0,1\}^n}$ with $m_{i,j}=\Delta(i,j)+2$ if $i\neq j$ and $m_{i,i}=0$ can not be embedded into the metric space $R=\{0,1\}^m$ for all $m<\frac{2^n-n-1}{2}$ with L^1 as distance-norm.*

Note that another intuitive argument for this statement is that the second part of the sum basically implies that the embedding contains a part which assigns to all 2^n bit-strings points such that their mutual distance is 2. This is called an equilateral embedding. Moreover, for a given metric space of dimension k the equilateral dimension is the maximal number of points which can be of mutually the same distance. [1] proved that for the integer lattice with L^1 norm, the equilateral dimension is $O(k\log k)$. Therefore the dimension in which this distance matrix can be embedded can not be much smaller than 2^n.

These tools are enough to provide a lower bound for the isometric dimension of Horn-encoded graphs.

Lemma 7. *For every n there is an induced graph G of the hypercube H_{2n+1} of size 2^n+2^{2n} with isometric dimension between $\frac{2^n-n-1}{2}$ and 2^n+2n which can be encoded in a $Horn_3$ formula of size $poly(n)$.*

Proof. Consider the formula $F(x_1,x_1',\ldots,x_n,x_n')=\bigwedge_{i\in[n]}(x_i\leftrightarrow x_i')$ and $F'=y\to F$ with a new variable y. Obviously $F'\in Horn_3$, F has a solution graph with 2^n isolated vertices and $G=G_{F'}$ is only a single connected component of size 2^n+2^{2n}. But by fixing y to 1 we get the original formula with 2^n isolated vertices u. All these vertices u agree on y but for $u\neq u'$, their distance in G is $\Delta(u,u')+2$. For all vertices u,u' with $y=0$, their distance is $\Delta(u,u')$.

So an isometric embedding for G implies an embedding for all u with the variable y set to 1. But by Corollary 2 we know that such an embedding needs at least $\frac{2^n-n-1}{2}$ bits. This proves our lower bound.

For an upper bound we replace y with 2^n bits and every node u of G which is isolated in G_F sets a different bit to 1. Every other node sets all new bits to 0. This is a correct partial cube embedding for F' of dimension 2^n+2n. □

For Horn-encoded graphs which are not even partial cubes, we provide an example:

Lemma 8. *There is a $Horn_4$ formula encoding a single connected component which is not a partial cube.*

Proof. Consider the formula $F(w, x, y, z) = (\overline{y} \to \overline{z}) \wedge ((w \wedge x) \to (y \leftrightarrow z))$. This clearly is a $Horn_4$ formula and the encoded graph is depicted in figure 2.

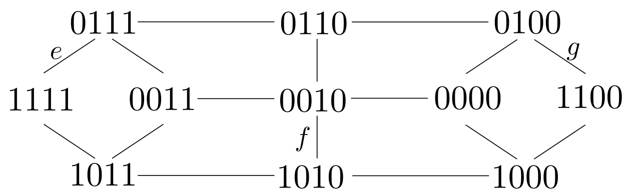

Figure 2: Induced subgraph of the Hypercube which is not a partial cube.

To see this, we note that a classical characterization of partial cubes is that an undirected graph $G = (V, E)$ is a partial cube iff it is bipartite and the relation Θ on the edge set is transitive. Hereby Θ is defined as $\{u, v\} \ \Theta \ \{x, y\} \leftrightarrow d(u, x) + d(v, y) \neq d(u, y) + d(v, x)$ for $\{u, v\}, \{x, y\} \in E$ (see for example [13]). It is easy to see that in the given example $e \ \Theta \ f$ and $f \ \Theta \ g$ but $e \not\Theta g$. Therefore the graph encoded by F is not a partial cube (no matter the isometric dimension). $\square$

We briefly mention an observation for length-bounded *st*-connectivity (*l-stConn(S)*). This problem is, given a formula F built from S, $s, t \models F$ and $l \in \mathbb{N}$, is there a path of length at most l from s to t? Clearly, if F is CPSS, then either $d(s, t) = \Delta(s, t)$ or $d(s, t) = \infty$. So *l-stConn(CPSS)* reduces to counting different bits and checking if there is a path at all. So for CPSS, this problem can be solved in P. In contrast, if F is Schaefer but not CPSS, this problem seems harder although the solution graph still has a small diameter. For *l-stConn($Horn_3$)*, [4] proved $W[2]$-hardness.

Theorem 1 ([4])**.** *l-stConn($Horn_3$) is $W[2]$-hard when parameterized by $l' = l - \Delta(s, t)$.*

5 Improved algorithms for connectivity and *st*-connectivity

In [17], the author gives a polynomial-time algorithm for connectivity on CPSS-formulas. But as this algorithm is basically a logspace-reduction to the satisfiability problem, we can refine this statement for more restricted classes of formulas. We restate this result and prove our corollary.

Theorem 2. *[17] Given a CPSS set S and a formula $F(x_1, \ldots, x_n)$ over S, the following polynomial-time algorithm decides whether F is connected: For every constraint C_i of F, obtain the projection F_i of F to the variables $\mathbf{x_i}$ occurring in C_i by checking for every assignment a of $\mathbf{x_i}$ whether $F[\mathbf{x_i}/a]$ is satisfiable. Then F is connected iff no F_i is disconnected.*[3]

Corollary 3. *For CPSS sets S, Conn(S) $\leq^L_T$ Sat(S).*

[3]Note that as S is finite, every constraint has finite arity and therefore a solution graph of constant size.

Lemma 9. *Conn(2SAT)* $\in$ *NL.*

Proof. It is easy to see that the solution graph of a satisfiable $2SAT$ formula is disconnected iff the implication graph contains a cycle. The proof of Lemma 10 gives more details on this statement. While checking for a cycle is in NL = coNL, checking if a solution graph is disconnected therefore is in coNL. It follows that *Conn(2SAT)* $\in$ NL. ☐

Now as Corollary 3 is a direct result of [17] in the case of *Conn*, the remaining part of this section will derive a similar statement for *stConn*. In addition, our following work will show that *Conn(2SAT)* is NL-complete and *Conn(*$Horn_3$*)* is P-complete.

In [8] the authors proved that *st*-connectivity is in P for all tight sets of functions by showing that the diameter in connected components is bounded by a linear function. We now show that even for tight formulas, *st*-connectivity can be reduced to a satisfiability problem.

Theorem 3. *Given a tight formula* $F(x_1, \ldots, x_n)$ *and* $s, t \in \{0,1\}^n$*. Then stConn(F,s,t)* $\leq_m^L$ $Sat(F \cup \{(x \vee \overline{y})\})$*.*

Proof. Given F as well as solutions s, t, we perform a walk on the solution graph starting at s by constructing a formula F' which is satisfiable iff there is a path from s to t in F.

We create a formula F' such that a satisfying assignment describes a walk from s to t in the solution graph. Various copies of the variables $x = (x_1, \ldots, x_n)$ of F simulate steps on the solution graphs. The first copy x^0 gets fixed to s. The additional copies can always only vary from the preceding copy in a specific variable and have to satisfy F. If the distance of consecutive copies x^i and x^{i+1} is at most one and the last copy is equal to t, then there is a path from s to t in F. If we know that s and t have distance at most d, we take dn steps by using d copies of the following construction. The set of variables x^1 is allowed to differ from x^0 only in the variable x_1. The next set of variables, x^2 can only differ from x^1 in x_2. After n such steps and d copies of this construction, we fix the last set of variables to t and know that the formula is satisfiable iff there is a path from s to t.

Note that in each step we only offer the new variable-set to flip a variable. We therefore fix all other variables to the previous value with clauses indicating equivalence and omit such clauses for the variable which is allowed to change. ☐

Corollary 4. *stConn(2SAT)* $\in$ *NL. stConn(S)* $\in$ *P for* S *a set of Schaefer functions.*

In addition, we prove the completeness results for *st*-connectivity problems on solution graphs of $2SAT$- and $Horn$-formulas.

Lemma 10. *stConn(2SAT) is NL-complete.*

Proof. We reduce the complement of the NL-complete problem of acyclic directed connectivity (see for example [2]) to *stConn(2SAT)*. The proof follows as NL = coNL.

Suppose we are given an acyclic directed graph $G = (V, E)$ and two nodes s, t. We now create G' by adding to G the edge (t, s). Clearly, there is a cycle in G' iff there is an s, t path in G. We now interpret G' as implication graph of a $2SAT$ formula F and state that there is a path from 0^n to 1^n in the solution graph G_F iff there is no cycle in G'.

Suppose there is no cycle. We describe a path from 0^n to 1^n. Every satisfying assignment coincides with a $0, 1$ labeling of G'. If a variable x_i is set to 0, then the node $x_i \in G'$ is labeled with 0. An assignment x is satisfying F as long as there is no edge (x_i, x_j) with $x_i = 1$ and $x_j = 0$. We therefore move from 0^n to 1^n by flipping the last bits of all longest path in G'. If all such bits were flipped, we delete the corresponding nodes and repeat until the graph is empty and we reach 1^n.

For the other direction, suppose there is a cycle. Then obviously we can not reach 1^n from 0^n. At one point we have to flip a single variable in this cycle. But then we would have to flip all variables in this cycle at the same time or else there would be an implication $1 \to 0$. So the graph is disconnected. □

Corollary 5. *Conn(2SAT) is NL-complete.*

Proof. This follows by the observation that in the previous reduction, the constructed solution graph is connected iff there is no cycle in the graph. As $2SAT$ is closed under complementation, our statement follows. For the upper bound, note that such a graph contains more than one connected component iff the associated implication graph contains a cycle. □

It would be interesting to give a direct NL algorithm for *stConn(2SAT)* using properties of the solution graph instead of just reducing to the satisfiability problem. This is still open. The main difference to the connectivity for CPSS formulas is that in the case of $2SAT$ formulas, the connected components are median graphs, a subclass of all partial cubes, while formulas which are CPSS consist of partial cubes of dimension n. Interestingly, the *st*-connectivity on $Horn_3$ formulas is complete for P while finding a shortest path is hard for $W[2]$ as explained in the next section.

Lemma 11. *stConn($Horn_3$) is P-complete.*

Proof. We use a similar method as in Lemma 10 and reduce the monotone circuit value problem to *stConn($Horn_3$)*. As the monotone circuit value problem is known to be P-complete [7], the hardness result follows.

So given a monotone circuit C with n inputs and bounded in-degree 2, m inner gates and an $x \in \{0, 1\}^n$, we first create a hypergraph G such that the marking algorithm[4] on hypergraphs reaches the root node iff $C(x) = 1$. $G = (V, E)$ with V is the set of gates in C and for a gate u with inputs v, w we add the edges

1. $((v, w) \to u)$ iff u is labeled with $\wedge$
2. and $(v \to u)$ and $(w \to u)$ iff u is labeled with $\vee$.

[4]This algorithm starts with a directed hypergraph and an initially marked set of nodes. If there is a hyperedge such that all source-nodes are marked but not all target nodes, we mark all target nodes. The algorithm finishes if there is no hyperedge which would mark a new node.

It is easy to see that if we mark all input nodes $x_i = 1$, the marking algorithm reaches the root z iff $C(x) = 1$. Note that the marking algorithm proceeds as follows: if there is an edge $((u_1, \dots, u_l) \to u)$ and all u_i are marked, we can mark u. We add some additional edges (z, x_i) for every $x_i = 1$. So the marking algorithm can perform a cycle in G iff $C(x) = 1$.

We now interpret this hypergraph as a Horn-formula F with $n+m$ variables and prove that the solution graph of F has a path from $1^{n+m} = x^*$ to 0^{n+m} iff $C(x) = 0$. Suppose first that $C(x) = 0$. We therefore know that the marking algorithm, starting with marked x_i for all $x_i = 1$ never reaches the root node z. Let $A(x)$ be the set of nodes this algorithm marks when starting with all 1 bits in x including exactly the variables which are initially set to 1. Then for all $u \notin A(x)$ without any predecessor in G, it is safe to flip $x^*[u]$ to 0. This corresponds to every input gate of C which is not set to 1. In a second round we flip all nodes with at least one predecessor which was already flipped (and where the premise is therefore false) to 0 and continue this process level by level until we reach the root z. Note that we never violate any clause of F. If we would violate a clause $((u, v) \to w)$ by setting the premise of the clause to 1 but the conclusion to 0, then $w \in A(x)$ which is a contradiction to the way we chose the variables.

We finish this process by reaching the root z. Now, in a second step, we can flip the input variables $x_i \in A(x)$ and perform the same process with all nodes in $A(x)$. This again does not violate any clauses and, in the end, $x^* = 0^{n+m}$.

Now suppose $C(x) = 1$ and $x^* = 1^n$. We just note there is no path in the solution graph of F to reach 0^n. We know that $|A(x)| \geq 2$ and for every single $u \in A(x)$, flipping u to 0 violates a clause in F. Any $u \in A(x)$ is the conclusion of a clause in F with the premise set to 1. So flipping any single variable in $A(x)$ violates F and there can not be a path from 1^{n+m} to 0^{n+m}. Note that this case did not occur for $C(x) = 0$ because we set z to 0 and then $A(x)$ had elements without predecessors in $A(x)$ (at first the inputs x_i). This finishes our proof. □

Corollary 6. *Conn(Horn$_3$) is hard for P.*[5]

Proof. This follows by the observation that in the previous reduction, the constructed solution graph is connected iff the marking algorithm does not reach the root node. As P is closed under complementation, our statement follows. □

A reduction from satisfiability of tight formulas to *st*-connectivity of tight formulas is not possible unless P = NP. To see this, we note that the work of [9] implies that satisfiability of tight formulas is NP-complete while *st*-connectivity is in P, see [17].

6 Conclusions and Open Problems

We have studied solution graphs of different sets of boolean formulas introduced by [8] and [17]. We showed that all solution graphs of CPSS formulas consist of partial cubes of small isometric dimension and by going to general Schaefer

[5]A previous version of this paper contained a typo which stated completeness for P which conflicts with the previously mentioned result that this problem is even coNP-complete (with an immensely more complex reduction).

formulas, their dimension may increase exponentially or they may even loose the property of being a partial cube. This gives a sharp separation between solution graphs which behave nicely and solution graphs without any known structure. It would be interesting to further analyze solution graphs of Horn formulas and either show that they behave still nice in another way or if they are already complicated enough for other problems to be much harder for these graphs. One of such problems is the connectivity problem as shown by [8, 17] which is coNP-complete. It would be interesting to find more such problems and further understand the origin of this complexity blow-up.

We introduced techniques to reduce connectivity and *st*-connectivity in CPSS or tight formulas to their satisfiability problem. We even proved the equivalence of these problems and satisfiability for related formulas. These results imply that for solution graphs of $2SAT$ formulas, a collection of undirected partial cubes, the *st*-connectivity problem is NL-complete while for Horn solution graphs it is P-complete. An explanation for this difference could be the fact that $2SAT$ formulas describe median graphs which are a proper subset of partial cubes. We would like to see an NL-algorithm for *stConn(2SAT)* which directly exploits this property. A similar statement holds for connectivity.

Simultaneously our results imply that length-bounded *st*-connectivity is easy for CPSS formulas while a result of Bonsma et al. [4] implied $W[2]$-hardness for general Schaefer-formulas. This implies that there is probably no polynomial-time reduction from *stConn(Horn)* to *stConn(CPSS)* which preserves distances.

References

[1] Noga Alon and Pavel Pudlák. Equilateral Sets in l_p^n. *Geometric And Functional Analysis*, 13(3):467–482, June 2003.

[2] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[3] José L. Balcázar, Antoni Lozano, and Jacobo Torán. The Complexity of Algorithmic Problems on Succinct Instances. *Computer Science, Research and Applications, Springer US*, 1992.

[4] Paul Bonsma, Amer E. Mouawad, Naomi Nishimura, and Venkatesh Raman. The Complexity of Bounded Length Graph Recoloring and CSP Reconfiguration. *Parameterized and Exact Computation, Lecture Notes in Computer Science*, 8894, 2014.

[5] Bireswar Das, Patrick Scharpfenecker, and Jacobo Torán. Succinct Encodings of Graph Isomorphism. In *Language and Automata Theory and Applications*, volume 8370 of *Lecture Notes in Computer Science*, pages 285–296. Springer International Publishing, 2014.

[6] Oya Ekin, Peter L. Hammer, and Alexander Kogan. On connected Boolean functions. *Discrete Applied Mathematics*, 96-97:337–362, October 1999.

[7] Leslie M. Goldschlager. The monotone and planar circuit value problems are log space complete for p. *SIGACT News*, 9(2):25–29, July 1977.

[8] Parikshit Gopalan, Phokion G. Kolaitis, Elitza Maneva, and Christos H. Papadimitriou. The Connectivity of Boolean Satisfiability: Computational and Structural Dichotomies. *SIAM Journal on Computing*, 38(6):2330–2355, January 2009.

[9] Laurent Juban. Dichotomy theorem for the generalized unique satisfiability problem. In Gabriel Ciobanu and Gheorghe Păun, editors, *Fundamentals of Computation Theory*, volume 1684 of *Lecture Notes in Computer Science*, pages 327–337. Springer Berlin Heidelberg, 1999.

[10] Elitza Maneva, Elchanan Mossel, and Martin J. Wainwright. A new look at survey propagation and its generalizations. *J. ACM*, 54(4), July 2007.

[11] Marc Mézard and Riccardo Zecchina. Random k-satisfiability problem: From an analytic solution to an efficient algorithm. *Phys. Rev. E*, 66:056126, Nov 2002.

[12] M. Mézard, F. Ricci-Tersenghi, and R. Zecchina. Two solutions to diluted p-spin models and xorsat problems. *Journal of Statistical Physics*, 111(3-4):505–533, 2003.

[13] Sergei Ovchinnikov. *Graphs and Cubes*. Universitext, Springer, 2011.

[14] Christos H. Papadimitriou and Mihalis Yannakakis. A note on succinct representations of graphs. *Information and Control*, 71(3):181–185, December 1986.

[15] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing - STOC '78*, pages 216–226, New York, New York, USA, May 1978. ACM Press.

[16] Patrick Scharpfenecker. On the structure of solution-graphs for boolean formulas. In *Fundamentals of Computation Theory - 20th International Symposium, FCT 2015, Gdańsk, Poland, August 17-19, 2015, Proceedings*, pages 118–130, 2015.

[17] Konrad W. Schwerdtfeger. A Computational Trichotomy for Connectivity of Boolean Satisfiability. page 24, December 2013.

[18] Helmut Veith. Languages represented by Boolean formulas. *Information Processing Letters*, 63(5):251–256, September 1997.

[19] Helmut Veith. How to encode a logical structure by an OBDD. In *Proceedings. 13th IEEE Conference on Computational Complexity*, pages 122–131. IEEE Comput. Soc, 1998.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de