The Complexity of Finding S-Factors in Regular Graphs

Sanjana Kolisetty
Departments of Mathematics and EECS, CSE Division,
University of Michigan, Ann Arbor, MI, USA
sanjanak@umich.edu

Linh Le
Departments of Mathematics and EECS, CSE Division,
University of Michigan, Ann Arbor, MI, USA
likle@umich.edu

Ilya Volkovich
Department of EECS, CSE Division, University of Michigan, Ann Arbor, MI, USA
ilyavol@umich.edu

Mihalis Yannakakis
Department of Computer Science, Columbia University, New York, NY, USA
mihalis@cs.columbia.edu

Abstract

A graph G has an *S-factor* if there exists a spanning subgraph F of G such that for all $v \in V : \deg_F(v) \in S$. The simplest example of such factor is a 1-factor, which corresponds to a perfect matching in a graph. In this paper we study the computational complexity of finding S-factors in regular graphs. Our techniques combine some classical as well as recent tools from graph theory.

2012 ACM Subject Classification Mathematics of computing → Matchings and factors; Theory of computation → Problems, reductions and completeness

Keywords and phrases constraint satisfaction problem, Dichotomy theorem, Graph Factors, Regular Graphs

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2019.21

Funding Research partially supported by the NSF-CCF AitF 1535912 Grant.
Sanjana Kolisetty: Research partially supported by the NSF-CCF AitF 1535912 Grant.
Linh Le: Research partially supported by the NSF-CCF AitF 1535912 Grant.

Acknowledgements The authors would like to thank the anonymous referees for their detailed comments and suggestions on the previous version of the paper.

1 Introduction

The *Constraint Satisfaction Problem* (CSP for short) has been a classical topic in computer science of both theoretical and practical importance. While CSPs can be quite general, in this paper we focus on the "fixed-template" Boolean CSPs. That is, CSPs over the Boolean domain where the constraints come from a fixed set of Boolean relations Γ. Formally, given a fixed set of Boolean relations $\Gamma = \{R_1, R_2, \ldots, R_m\}$, a Γ-*formula* is a conjunction of constraints of the form $R_j(x_{i_1}, \ldots, x_{i_n})$ where $R_j \in \Gamma$ and the x_{i_j}-s are propositional variables; CSP(Γ) forms a decision problem where one needs to determine if a given Γ-formula is satisfiable. In other words, one needs to determine whether it is possible to satisfy all the constraints as given by the relations from Γ simultaneously.

39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019).
Editors: Arkadev Chattopadhyay and Paul Gastin; Article No. 21; pp. 21:1–21:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The object of study is the computational complexity of CSP(Γ) as per the choice of Γ. In a seminal work of [22], Schaefer identified six classes of sets of Boolean relations for which CSP(Γ) ∈ P and proved that all other sets of relations generate an NP-complete problem. This result is what is known as **Schaefer's Dichotomy Theorem** which provides a complete classification of the computational complexity of CSP(Γ). The two most popular examples of applications of this theorem are the NP-completeness of the 1-*in*-3SAT and *not-all-equal* 3SAT (NAE-3SAT) problems. Subsequently, in [3], a more refined classification was presented.

While a more general Dichotomy Theorem was recently proved for non-Boolean CSPs [6, 24][1], there has been a large body of work dedicated to the study of the computational complexity of a restricted version of CSP(Γ), denoted as $\mathrm{CSP}_2(\Gamma)$ or $\mathrm{CSP}_{\mathrm{Edge}}(\Gamma)$ [15, 10, 7, 14, 11, 8, 17]. Formally introduced by Feder in [10], $\mathrm{CSP}_2(\Gamma)$ corresponds to a specialization of CSP(Γ) to the instances where each variable appears at most twice. Alternatively, one can think about embedding the input Γ-formula into a graph, such that edges correspond to variables and nodes to constraints, and the constraint satisfaction problem asks for a spanning subgraph such that the set of its edges at each node satisfies the constraint at the node.

The main subtlety is that if CSP(Γ) ∈ P then, clearly, $\mathrm{CSP}_2(\Gamma) \in \mathsf{P}$. However, in the general NP-hard instances there is usually no restriction of the number of appearances of a variable. Therefore, a proof that CSP(Γ) is NP-hard may not carry over to $\mathrm{CSP}_2(\Gamma)$. In particular, if we consider the aforementioned examples, CSP_2(1-in-3SAT) corresponds to determining existence of a perfect matching in a 3-regular graph, which is decidable in polynomial time. In addition, CSP_2(NAE-3SAT) is "trivial" since every read-twice[2] NAE-3SAT-formula is always satisfiable![3]

Despite all the invested effort, we are still far away from the ultimate goal. Indeed, the known results do not even provide a complete classification for the cases when Γ consists of just a single relation! A natural focus taken in [15], was to consider the sets Γ that consist of symmetric relations as these instances often arise more naturally in the graph context, because incident edges to a node are typically treated symmetrically in graph theory. This class of problems can be regarded as generalized matchings. In this paper we give a complete classification of the computational complexity of $\mathrm{CSP}_2(\Gamma)$, where Γ consists of a single symmetric relation.

In turns out that the problem has a very natural interpretation in terms of finding "S-factors" of regular graphs. We say that a graph G has an *S-factor*, if there exists a spanning subgraph F of G such that for all $v \in V : \deg_F(v) \in S$. The simplest example of such factor is a 1-factor, which corresponds to a perfect matching in a graph.

1.1 Results

In light of the natural interpretation of the problem in terms of finding S-factors of graphs, we present our main results using that terminology. The CSP versions of the main results (and their proofs) can be found in Section 4. Our first result is a Dichotomy Theorem for regular graphs of even degree. The Dichotomy is obtained by classifying all the tractable cases.

---

1. Affirming what was known as the CSP Dichotomy Conjecture formulated in [12].
2. A formula in which every variable appears at most twice.
3. This follows immediately from Tutte's Theorem (Lemma 8), however there is a more direct way to see that.

▶ **Theorem 1.** *Let $\ell \in \mathbb{N}$. There is a polynomial-time algorithm that given a 2ℓ-regular graph G as an input, finds an S-factor in G, if there is one, in the following four cases:*
1. *S contains an even number.*
2. *$\ell \in S$.*
3. *$\{\ell - 1, \ell + 1\} \subseteq S$.*
4. *$S = \{p, p + 2, \cdots, p + 2r\}$ for some $p, r \geq 0$.*

Otherwise, finding an S-factor is NP*-Hard.*

As could be observed, all the tractable cases reduce to the case of finding a perfect matching in a graph (see Section 2.1 for details). Consequently, an algorithm for perfect matching in graphs (e.g. [9]) could be used to find these S-factors, for the "yes"-instances of the problem. For regular graph of odd degree, we obtain a somewhat weaker result: we show that for each set S, the decision problem is either polynomial-time solvable or NP-hard, yet we are unable to classify explicitly all the tractable cases. Closing this gap will require resolving several conjectures in graph theory (see [2, 19, 1, 5] for more details).

▶ **Theorem 2.** *Let $\ell \in \mathbb{N}$. There is a polynomial-time algorithm that given a $(2\ell + 1)$-regular graph G as an input, decides if G has an S-factor, in the following two cases:*
1. *Every $(2\ell + 1)$-regular graph has an S factor.*
2. *$S = \{p, p + 2, \cdots, p + 2r\}$ for some $p, r \geq 0$.*

Otherwise, deciding if G has an S-factor is NP*-Hard.*

There are specific sets S, for which it is an open problem in graph theory whether every $(2\ell + 1)$-regular graph has an S-factor. A simple concrete example is the case of $S = \{1, 4\}$ for degree-5 graphs (the conjecture in this case is that there is always an S-factor). The theorem tells us that, even though we may not know the answer to the open problem for a particular S, if it does not hold trivially for all graphs and there is a counterexample, then the corresponding S-factor problem is NP-hard; that is, there is a way to use any counterexample (as a black box) to generate an NP-hardness reduction.

1.2 Comparison to Previous Results

In [15], Istrate studied the special case when Γ consists of symmetric relations. In that work, several "patterns" for which $\mathrm{CSP}_2(\Gamma) \in \mathsf{P}$ were identified. In particular, one such pattern corresponds to Case 4 of Theorem 1. This result was obtained via connections to covering problems. In addition, Istrate formulated a sufficient condition under which the computational complexity of $\mathrm{CSP}_2(\Gamma)$ and $\mathrm{CSP}(\Gamma)$ is the same, with the additional "constants for free" assumption. That is, one can fix some variables to either 0 and 1 (for more details, see Lemma 26 and the preceding discussion). Later on, Feder [10], extended the condition to non-symmetric relations, introducing *Delta Matroids*, and showed that if Γ contains some relation that is not a Delta matroid then $\mathrm{CSP}_2(\Gamma)$ and $\mathrm{CSP}(\Gamma)$ have the same complexity (in the presence of constants). Several subsequent works [7, 8, 17] introduced further refinements to Delta Matroids. Yet, "constants for free" remained a prevalent assumption in these and other CSP-related works. Nonetheless, even with the assumption, no classification for the mere case of a single symmetric relation was known prior to our work.

We also would like to point out that the "constants for free" assumption is implicitly equivalent to adding two more relations $P(x) = x$ and $Q(x) = \neg x$ to Γ. It is important to stress that adding these relations can completely tilt the scale. For example, consider a single 8-ary symmetric relation "two or six out of eight". Formally, $R(\bar{x}) = 1$ iff $w_H(x) = 2$ or 6. In the graphical perspective, this corresponds to the problem of finding a $\{2, 6\}$-factor of an 8-regular graph. Now by TutteŠs Theorem (Lemma 8), every 8-regular graph has a 2-factor. Hence $\mathrm{CSP}_2(R) \in \mathsf{P}$ in a "trivial" way. On the other hand, $\mathrm{CSP}_2(\{R, x, \neg x\})$ is NP-hard (follows e.g. from [15]). Our results do not rely on the "constants for free" assumption. In fact, they complement it: roughly speaking, we show that either $\mathrm{CSP}_2(\Gamma) \in \mathsf{P}$ or there exist Γ-formulas that "implement" the relations x and $\neg x$. See Lemmas 29 and 30 for more details.

There is, of course, extensive work in graph theory on factors in graphs, (see e.g. the surveys [2, 21] and references therein), with the development of a rich theory of matchings, as well as more general factors. This includes structural results on the existence of factors, starting from Petersen's theorem from 1891 [20]; algorithmic results, including e.g. Edmonds' matching algorithm [9] and its extensions and refinements; and hardness results, starting e.g. with Lovász's theorem [18] that for any $a, b \in \mathbb{N}$ such that $1 \le a \le b-3$, the problem of deciding whether a graph has an $\{a, b\}$-factor is NP-hard even for simple graphs (not necessarily of a given, regular degree). We will leverage several of these graph theoretic results on (generalized) matchings and the existence of suitable factors in graphs. We review some of these theorems that we use in the next section.

2 Preliminaries

▶ **Definition 1** (Zebras and Holes). *Let $S \subseteq \mathbb{N}$ be a subset of $\mathbb{N}$. Following [15], we say that S contains a* hole of size t *if there exist i such that: $i, i+t+1 \in S$ and $[i+1, i+t] \cap S = \emptyset$. Let $a \le b \in \mathbb{N}$ such that $a \equiv b (\bmod 2)$. We say that S is an* (a, b)-zebra *if $S = \{a, a+2, a+4, \ldots, b\}$. We call a set S a zebra, if it is an (a, b)-zebra for some $a, b \in \mathbb{N}$.*

▶ Remark. A set $S = \{a\}$ also constitutes a zebra since it is an (a, a)-zebra. The following is a simple observation about the structure of finite subsets of $\mathbb{N}$, that will be useful for us later.

▶ **Observation 2.** *Let $S \subseteq \mathbb{N}$ be a finite, non-empty subset of $\mathbb{N}$ then (at least) one of the following holds:*
- *S contains two consecutive numbers.*
- *S is zebra.*
- *S contains a hole of size at least 2.*

2.1 Graphs

In this paper we consider graphs $G = (V, E)$. Unless specified otherwise, all the graphs considered in the paper are general graphs (i.e. with self-loops and parallel edges). The focus of this paper is the complexity of finding a particular kinds of subgraphs in graphs, known as *factors*. We define this formally now.

▶ **Definition 3** (Factors). *Let $G = (V, E)$ be a graph with V vertices and E edges. [2]*
1. *H-factor: Let H be a set function associated with G that maps $V \to 2^{\mathbb{N}}$. We say that G has an H-*factor *if there exists a spanning subgraph F of G such that for all $v \in V : \deg_F(v) \in H(v)$.*
2. *f-factor is a specialization to the case when $\forall v \in V : H(v) = \{f(v)\}$, for some function $f : V(G) \to \mathbb{Z}_+$.*
3. *S-factor is a specialization to the case when $H(v) = S$ for all $v \in V$, for some* fixed *set $S \subseteq \mathbb{N}$.*
4. *$[a, b]$-factor is a further specialization to the case when S is the interval $[a, b]$.*
5. *k-factor is a further specialization to the case when $S = \{k\}$.*

The simplest case of a graph factor is a 1-factor which corresponds to a perfect matching of a graph. This problem has a well-known efficient algorithm known as "Blossom Algorithm".

▶ **Lemma 4** ([9]). *There exists a polynomial-time algorithm that given a graph $G = (V, E)$ as an input, outputs a 1-factor F of G, if one exists.*

The algorithm can be easily extended to handle f-factors due to the following observation:

▶ **Lemma 5** ([4]). *There exists a polynomial-time algorithm that given a graph $G = (V, E)$ and a function $f : V \to \mathbb{Z}$ (as a vector) as an input, outputs a graph G' such that G' has a 1-factor F' iff G has an f-factor F. Moreover, F can be computed in polynomial time given F'.*

In addition, using the simple idea of [16] of introducing self-loops, the algorithm can be further extended to H-factors, where each $H(v)$ is a zebra (See Definition 1).

▶ **Lemma 6** ([16]). *There exists a polynomial-time algorithm that given a graph $G = (V, E)$ and a function $H : V \to 2^{\mathbb{N}}$, where each $H(v)$ is zebra, as an input, outputs a graph G' and a function $f : V \to \mathbb{Z}$ such that G' has a f-factor F' iff G has an H-factor F. Moreover, F can be computed in polynomial time given F'.*

The following is immediate given the above reductions to the perfect matching case.

▶ **Corollary 7.** *There exists a polynomial-time algorithm that given a graph $G = (V, E)$ and a function $H : V \to 2^{\mathbb{N}}$, where each $H(v)$ is zebra, as an input, outputs an H-factor F of G, if one exists.*

We note that an efficient algorithm for this kind of H-factors has been obtained in [15] using a different argument. Recently in [17], the algorithm was extended to also handle the "asymmetric" version. Next, we require the following results regarding the existence of regular factors in regular graphs.

▶ **Lemma 8** (Regular Factors of Regular Graphs).

1. [23] *Let $r, k \in \mathbb{N}$ such that $1 \leq k \leq r-1$. Then any r-regular graph has a $\{k, k+1\}$-factor.*
2. [20] *Let r and k be even integers such that $2 \leq k \leq r$. Then any r-regular graph has a k-factor.*
3. [13] *Suppose r is even and $\frac{r}{2}$ is odd. Then any connected r-regular graph of even order has a $\frac{r}{2}$-factor.*

As a corollary we obtain the following, which was observed for simple graphs in [1]. We also note that the proof of [1] is merely existential whereas our proof is algorithmic.

▶ **Lemma 9.** *Let $r \in \mathbb{N}$ such that both r and $\frac{r}{2}$ are even. Then any r-regular graph of even order has a $\left\{\frac{r}{2} - 1, \frac{r}{2} + 1\right\}$-factor.*

Proof. Let $G = (V, E)$ be a graph satisfying the preconditions. For every $v \in V$, we add a self-loop. Call this new resulting graph $G' = (V, E')$. Observe that G' is a $(r+2)$-regular graph of even order and $\frac{r+2}{2} = \frac{r}{2} + 1$ is odd. Therefore, by Lemma 8, G' has a $(\frac{r}{2} + 1)$-factor. Now, consider two cases: if $v \in V$ uses the self-loop to fulfill its factor, then the induced degree of v in G is $\frac{r}{2} - 1$. Otherwise, the induced degree of v in G is $\frac{r}{2} + 1$. ◀

2.2 Boolean Relations

▶ **Definition 10.** *The* Hamming Weight *of a vector $\bar{v} \in \{0, 1\}^n$ is defined as:* $\mathrm{w_H}(\bar{v}) \triangleq |\{i \mid v_i \neq 0\}|$. *That is, the number of its non-zero coordinates.*

▶ **Definition 11** (Symmetric Relation). *We say that an m-ary relation $R(x_1, x_2, \dots, x_m)$ is* symmetric *if there exists a set* $\mathsf{Spec}(R) \subseteq \{0 \dots m\}$ *such that* $R(\bar{x}) = 1$ *if and only if* $\mathrm{w_H}(\bar{x}) \in \mathsf{Spec}(R)$. *The set* $\mathsf{Spec}(R)$ *is called the* spectrum *of* R.

The following are examples of particular symmetric relations we will be utilizing.

▶ **Example 12.**

- $\mathrm{NE}(x_1, x_2)$ is a binary relation with $\mathsf{Spec}(\mathrm{NE}) = \{1\}$.
- Let $k \in \mathbb{N}$. EQ_k is a k-ary relation with $\mathsf{Spec}(\mathrm{EQ}_k) = \{0, k\}$.

▶ **Definition 13** (Dual Relation). *Let $R(x_1, \dots, x_m)$ be a relation. We define the* dual relation *of R as:*

$$R^*(x_1, \dots, x_m) \triangleq R(\neg x_1, \neg x_2, \dots, \neg x_m).$$

The following observation is immediate with respect to symmetric relations.

▶ **Observation 14.** *For a symmetric m-ary relation R we have:* $\mathsf{Spec}(R^*) = \{m - i \mid i \in \mathsf{Spec}(R)\}$.

2.2.1 Γ-Instances, $\mathrm{CSP}(\Gamma)$, Triviality

In what follows, let $\Gamma = \{R_1, R_2, \dots, R_\ell\}$ be a *fixed* set of Boolean relations. We will use Γ^* to denote the set of dual relations. Formally, $\Gamma^* \triangleq \{R_1^*, R_2^*, \dots, R_\ell^*\}$, where R_i^* is the dual relation of R_i.

▶ **Definition 15.** *A* Γ-instance *or* Γ-formula Φ *is a conjunction of constraints of the form $R_j(x_{i_1}, \dots, x_{i_n})$ where $R_j \in \Gamma$ and the x_{i_j}-s are propositional variables. The* read *of a variable x_i in Φ is the number of occurrences of x_i in Φ. The* read *of a formula Φ is the maximal read of a variable in it.*

In this paper we will focus on read-twice formulas, that is formulas in which all the variables appear at most two times. We now formally introduce the main problem we will study.

▶ **Problem 16.** $\mathrm{CSP}(\Gamma)$ *forms a decision problem where one needs to determine if a given Γ-formula is satisfiable. In other words, one needs to determine whether it is possible to satisfy **all** the constraints as given by the relations from Γ, simultaneously. For $k \geq 1$,* $\mathrm{CSP}_k(\Gamma)$ *is a specialization of* $\mathrm{CSP}(\Gamma)$ *to read-k instances. If $\Gamma = \{R\}$ has a single relation R, we will write* $\mathrm{CSP}(R)$ *and* $\mathrm{CSP}_k(R)$.

As was pointed out in the introduction, in this paper we are interested in the computational complexity of $\mathrm{CSP}_2(\Gamma)$, as per the choice of Γ. We now recall Schaefer's Dichotomy Theorem [22].

▶ **Lemma 17** ([22]). $\mathrm{CSP}(\Gamma) \in \mathsf{P}$ *in the following six cases:*

1. $\forall R_j \in \Gamma : R_j(\bar{0}) = 1$
2. $\forall R_j \in \Gamma : R_j(\bar{1}) = 1$
3. $\forall R_j \in \Gamma : R_j$ *is equivalent to a conjunction of binary relations*
4. $\forall R_j \in \Gamma : R_j$ *is equivalent to a conjunction of Horn clauses*
5. $\forall R_j \in \Gamma : R_j$ *is equivalent to a conjunction of dual-Horn clauses*
6. $\forall R_j \in \Gamma : R_j$ *is equivalent to a conjunction of affine forms*

Otherwise, $\mathrm{CSP}(\Gamma)$ *is* NP*-Hard.*

The following is an instantiation of the Theorem to the case of a single symmetric relation.

▶ **Corollary 18.** *Let $R(x_1, \ldots, x_m)$ be a symmetric relation. Then* $\mathrm{CSP}(R) \in \mathsf{P}$ *in the following cases:*
1. $R(\bar{0}) = 1$
2. $R(\bar{1}) = 1$
3. $m \leq 2$
4. $\mathsf{Spec}(R)$ *contains all odd numbers in* $\{1, \ldots, m\}$

▶ **Definition 19** (Triviality). *We say that* $\mathrm{CSP}_k(\Gamma)$ *is* trivial *if every instance of* $\mathrm{CSP}_k(\Gamma)$ *(i.e. every read-k Γ-instance) is satisfiable.*

To put the above definitions into a context, observe the first two of the six tractable classes correspond to cases when $\mathrm{CSP}(\Gamma)$ and $\mathrm{CSP}(\Gamma^*)$ are trivial. Similarly, observe that Cases 4 and 5 correspond the same conditions applied to both $\mathrm{CSP}(\Gamma)$ and $\mathrm{CSP}(\Gamma^*)$. With some extra work, you can see that the same holds true for Cases 3 and 6. In the same vein, the following lemma is immediate from the definition.

▶ **Lemma 20.** 1. $\mathrm{CSP}_1(\Gamma)$ *is trivial, as long as Γ does not contain a contradiction.*
2. *For any $k \in \mathbb{N}$ and any Γ:* $\mathrm{CSP}_k(\Gamma)$ *is trivial iff* $\mathrm{CSP}_k(\Gamma^*)$ *is trivial.*

2.2.2 Induced Relations and Implementation

▶ **Definition 21** (Induced relation). *For a relation $R(\bar{x}, \bar{y})$ we define the* induced relation *$\exists \bar{y} R(\bar{x}, \bar{y})$ on $\bar{x}$ as*

$$\exists \bar{y} R(\bar{x}, \bar{y}) = 1 \iff \exists \bar{y} \text{ such that } R(\bar{x}, \bar{y}) = 1.$$

▶ **Definition 22** (Implementation). *Let $R(\bar{x})$ be an arbitrary relation. We say that Γ* implements *R, denoted by Γ imp R, if there exists a Γ-instance $\Phi(\bar{x}, \bar{y})$ such that $R(\bar{x}) = \exists \bar{y} \Phi(\bar{x}, \bar{y})$. Furthermore, we say that Γ* read-twice-implements *R, denoted by Γ imp_2 R, if in addition:*
1. *Each x_i is read-once in Φ.*
2. *Each y_j is (at most) read-twice in Φ.*

The intuition behind the definition is that if Γ read-twice-implements R then we can, effectively, consider the set $\Gamma \cup \{R\}$ instead of Γ. The following lemma summarizes this intuition and will be used implicitly in our proofs.

▶ **Lemma 23.** *Let R be a relation such that Γ read-twice-implements R and let $\Gamma' \triangleq \Gamma \cup \{R\}$. Then for every read-twice Γ'-instance $\Phi'(\bar{x})$ there exists a read-twice Γ-instance $\Phi(\bar{x}, \bar{y})$ such that $\Phi'(\bar{x}) = \exists \bar{y} \Phi(\bar{x}, \bar{y})$.*

The following lemma showcases this intuition further, by showing that a three-way Equality EQ_3 can be used to implement k-way equality EQ_k for any $k \geq 3$. Conversely, one can use k-way equality EQ_k to implement k'-way equality $\mathrm{EQ}_{k'}$ for any $k' \leq k$.

▶ **Lemma 24.** *If Γ read-twice-implements EQ_3 then Γ read-twice-implements EQ_k, for any $k \geq 3$.*

Proof. By induction on k. The base case $k = 3$ is trivial. Let Φ_k denote a Γ-instance that read-twice implements $\mathrm{EQ}_k(x_1, \ldots, x_k)$. Given Φ_k, we can read-twice implement $\mathrm{EQ}_{k+1}(x_1, \ldots, x_{k+1})$ in the following way:

$$\Phi_k(x_1, \ldots, x_{k-1}, y) \wedge \Phi_3(y, x_k, x_{k+1}).$$

Given our inductive hypothesis and the fact that we can read-twice implement EQ_3, we can conclude that Γ read-twice-implements EQ_{k+1}. ◀

We note this was already observed in [15, 10]. Similar ideas can be used to show that if Γ read-twice-implements particular relations, then read-twice Γ-formulas exhibit some interesting closure properties.

▶ **Lemma 25** (Read-Twice Implementing Particular Relations)**.**

1. ***Closure Under Variable Negation:*** *Suppose* Γ *read-twice-implements* NE*. Then read-twice* Γ*-formulas are closed under variable negation. Formally, if* $\Gamma \text{ imp}_2 R(x, \bar{y})$ *then* $\Gamma \text{ imp}_2 R(\neg x, \bar{y})$*.*
2. ***Closure Under Setting Variables to Constants:*** *Suppose* Γ *read-twice-implements* x *or* $\neg x$*. Then read-twice* Γ*-formulas are closed under setting variables to either* 1 *or* 0*, respectively. Formally, if* $\Gamma \text{ imp}_2 R(x, \bar{y})$ *then* $\Gamma \text{ imp}_2 R(1, \bar{y})$ *or* $R(0, \bar{y})$*, respectively.*
3. ***Closure Under Variable Repetition:*** *Suppose* Γ *read-twice-implements* EQ_k*. Then read-twice* Γ*-formulas are closed under repetition of any variable arbitrary number of times.*

We will use the above implicitly. We finish this section with the following simple observation from [15].

▶ **Lemma 26** ([15])**.** *Let* R *be a symmetric relation such that* $\mathsf{Spec}(R)$ *contains a hole of size at least* 2*. Then* $\{R, x, \neg x\}$ *read-twice-implements* EQ_3*.*

We note that Feder [10] extended this claim to a non-symmetric case defining *Delta Matroids*. In the same paper it was observed that WLOG every variable in a read-twice formula occurs *exactly* twice. Furthermore, such formulas have very natural interpretation as graphs where edges play the role of variables and nodes the role of constraints.

▶ **Lemma 27** (Graphical Perspective of CSP_2 [10])**.** *Every read-twice formula can be efficiently transformed into an exact read-twice formula, and furthermore viewed as a graph.*

3 Main Technical Tools

In this section we present our main technical tools, which we will use to prove Theorems 1 and 2. We begin by showing that the sets Γ for which $\mathrm{CSP}_2(\Gamma)$ is non-trivial (in the sense of Definition 19), read-twice implement $\mathrm{NE}(x, y)$ or $\{x, \neg x\}$. Consequently, by Lemma 25, such read-twice Γ-formula are closed under variable negation or setting variables to constants $\{0, 1\}$. Note that the result holds for general relations (not necessarily symmetric).

▶ **Lemma 28.** *Suppose that* $\mathrm{CSP}_2(\Gamma)$ *is non-trivial. Then* Γ *read-twice-implements* $\mathrm{NE}(x, y)$ *or* $\{x, \neg x\}$*.*

Proof. Since $\mathrm{CSP}_2(\Gamma)$ is non-trivial, there exists an unsatisfiable read-twice Γ-instance Φ. On the other hand, recall (e.g. Lemma 20) that any read-once Γ-instance is satisfiable. Consider the "unpaired" version Φ' of Φ. Formally, for each variable x_i we replace one of the occurrences with a fresh new variable y_i. Observe that the resulting Φ' is read-once and hence satisfiable. Now, consider the process of gradually pairing the variables of Φ', that will eventually recover Φ. Formally, $\Phi'_0 \triangleq \Phi'$. Φ'_1 results from Φ'_0 by setting $x_1 = y_1$. More generally, Φ'_i results from Φ'_{i-1} by setting $x_i = y_i$. As $\Phi'_0 = \Phi'$ is satisfiable and $\Phi'_n = \Phi$ is not, by a hybrid argument, there exist i such that Φ'_{i-1} is satisfiable and Φ'_i is not. Let $\varphi(x_i, y_i)$ be the relation given by Φ'_{i-1} induced to the variables x_i and y_i (Recall Definition 21). By the above, $\varphi(0, 0) = \varphi(1, 1) = 0$ and either $\varphi(0, 1) = 1$ or $\varphi(1, 0) = 1$ (or both). Consider three cases:

- $\varphi(0,1) = \varphi(1,0) = 1$. In this case: $\varphi(x_i, y_i) = \mathrm{NE}(x_i, y_i)$.
- $\varphi(0,1) = 0, \varphi(1,0) = 1$. In this case: $\exists y_i \varphi(x_i, y_i) = x_i$ and $\exists x_i \varphi(x_i, y_i) = \neg y_i$.
- $\varphi(0,1) = 1, \varphi(1,0) = 0$. In this case: $\exists y_i \varphi(x_i, y_i) = \neg x_i$ and $\exists x_i \varphi(x_i, y_i) = y_i$. ◀

Next, we show that for symmetric relations we can derive further closure properties under some technical conditions.

▶ **Lemma 29.** *Let R be a symmetric 2ℓ-ary relation such that: $\ell \notin \mathsf{Spec}(R)$ and $\{\ell-1, \ell+1\} \not\subseteq \mathsf{Spec}(R)$. Then $\{R, \mathrm{NE}\}$ read-twice-implements EQ_3 or $\{x, \neg x\}$.*

Proof. We define the following two sets: $S_- = \{a \mid R(\ell - a) = 1\}$ and $S_+ = \{a \mid R(\ell + a) = 1\}$. Furthermore, let $a_- = \min S_-$ and $a_+ = \min S_+$. We define a_- or a_+ to be infinity if S_- or S_+ is empty, respectively. We consider three cases:

- **Case 1:** $a_+ = a_-$. Observe that $a_- \geq 2$. Using NE and Lemma 25, we plug $\ell - a_-$ pairs $z_i, \neg z_i$ into the relation R. Formally, consider,

  $$R(\bar{z}, \bar{y}) \triangleq R(z_1, \neg z_1, \ldots, z_{\ell-a_-}, \neg z_{\ell-a_-}, y_1, \ldots, y_{2a_-}).$$

  By definition, $\mathrm{w_H}(\bar{z}) = \ell - a_-$ and $0 \leq \mathrm{w_H}(\bar{y}) \leq 2a_-$. Now, since $a_+ = a_-$:

  $$R(\bar{z}, \bar{y}) = 1 \iff \mathrm{w_H}(\bar{y}) \in \{0, 2a_-\}.$$

  Consequently, $\exists \bar{z} R(\bar{z}, \bar{y}) = \mathrm{EQ}_k(\bar{y})$, where $k = 2a_- \geq 4$.
- **Case 2:** $a_+ > a_-$. Observe that $a_- \geq 1$ and consider $R(\bar{z}, \bar{y})$ as above. Now, however, since $a_+ > a_-$:

  $$R(\bar{z}, \bar{y}) = 1 \iff \mathrm{w_H}(\bar{y}) = 0.$$

  Hence, we obtain $\neg y_i$. Using NE, we can obtain y_i.
- **Case 3:** $a_- > a_+$. Observe that $a_+ \geq 1$. We repeat the argument of Case 2 for the dual relation R^* of R. As R^* read-twice-implements $\{x, \neg x\}$, so does R. ◀

We use a similar argument for relations of odd arity.

▶ **Lemma 30.** *Let R be a symmetric $2\ell+1$-ary relation such that: $\{\ell, \ell+1\} \not\subseteq \mathsf{Spec}(R)$. Then $\{R, \mathrm{NE}\}$ read-twice-implements EQ_3 or $\{x, \neg x\}$.*

Proof. We define the following two sets: $S_- = \{a \mid R(\ell - a) = 1\}$ and $S_+ = \{a \mid R(\ell + 1 + a) = 1\}$. Furthermore, let $a_- = \min S_-$ and $a_+ = \min S_+$. We define a_- or a_+ to be infinity if S_- or S_+ is empty, respectively. We consider three cases:

- **Case 1:** $a_+ = a_-$. Observe that $a_- \geq 1$. Consider,

  $$R(\bar{z}, \bar{y}) \triangleq R(z_1, \neg z_1, \ldots, z_{\ell-a_-}, \neg z_{\ell-a_-}, y_1, \ldots, y_{2a_-+1}).$$

  By definition, $\mathrm{w_H}(\bar{z}) = \ell - a_-$ and $0 \leq \mathrm{w_H}(\bar{y}) \leq 2a_- + 1$. Now, since $a_+ = a_-$:

  $$R(\bar{z}, \bar{y}) = 1 \iff \mathrm{w_H}(\bar{y}) \in \{0, 2a_- + 1\}.$$

  Consequently, $\exists \bar{z} R(\bar{z}, \bar{y}) = \mathrm{EQ}_k(\bar{y})$, where $k = 2a_- + 1 \geq 3$.
- **Case 2:** $a_+ > a_-$. Observe that $a_- \geq 0$ and consider $R(\bar{z}, \bar{y})$ as above. Now, however, since $a_+ > a_-$:

  $$R(\bar{z}, \bar{y}) = 1 \iff \mathrm{w_H}(\bar{y}) = 0.$$

  Hence, we obtain $\neg y_i$. Using NE, we can obtain y_i.
- **Case 3:** $a_- > a_+$. Observe that $a_+ \geq 0$. We repeat the argument of Case 2 for the dual relation R^* of R. As R^* read-twice-implements $\{x, \neg x\}$, so does R. ◀

4 Characterization Proof

In this sections we give our main results, thus proving Theorems 1 and 2.

▶ **Theorem 31** (Characterization of Even-Arity Relations). *Let R be a symmetric 2ℓ-ary relation which is not constantly false. Then $\mathrm{CSP}_2(R) \in \mathsf{P}$ in the following four cases:*

1. *There is an even $k \in \mathsf{Spec}(R)$.*
2. *$\ell \in \mathsf{Spec}(R)$.*
3. *$\{\ell-1, \ell+1\} \subseteq \mathsf{Spec}(R)$.*
4. *$\mathsf{Spec}(R)$ is a zebra.*

Otherwise, $\mathrm{CSP}_2(R)$ is NP-Hard.

Proof. For Cases 1–4, we will take the graphical perspective (Lemma 27). Indeed, the problem corresponds to finding an S-factor of a given 2ℓ-regular graph, where $S = \mathsf{Spec}(R)$.

1. Follows from Item 2 of Lemma 8.
2. We can assume WLOG that S contains only odd numbers. In particular, ℓ is odd. Consider the following algorithm, given a graph G as an input.
   - Find all the connected components $C_1, C_2, \ldots, C_t$ of G.
   - If each C_i is of even order, return “true”; otherwise, return “false”.

   Analysis: If each C_i is of even order, then by Lemma 8, each C_i has an ℓ-factor and so does G. Conversely, suppose some C_i is of odd order. Then by Handshaking Lemma, C_i cannot have an S-factor, as otherwise the overall sum of the degrees will be odd.
3. As before, we can assume WLOG that S contains only odd numbers. Hence, ℓ is even. Apply the procedure outlined in the proof of Lemma 9. This will reduce the problem to the previous case.
4. Apply Corollary 7 with $H(v) = \mathsf{Spec}(R)$ for every vertex v in the graph.

For the NP-Hardness proof, we take the CSP view of the problem. We show that if none of the Cases 1-4 hold, then $\mathrm{CSP}_2(R)$ is as hard as $\mathrm{CSP}(R)$. That is, we can lift the restriction on the read. The hardness then follows from Schaefer’s Dichotomy Theorem instantiated to a single symmetric relation - Corollary 18.

▷ Claim 32. If $\mathsf{Spec}(R)$ does not fall into any of the four cases, then $\mathsf{Spec}(R)$ contains a hole of size at least 2 and $\{R\}$ read-twice-implements EQ_3.

Proof. First, observe that $\mathsf{Spec}(R)$ cannot have two consecutive numbers (as one of them will be even) and is not a zebra (Case 4). Therefore, by Observation 2, $\mathsf{Spec}(R)$ must contain a hole of size at least 2.
Next, consider the relation:

$$N(x, y) \triangleq \exists \bar{z} R(z_1, z_1, z_2, z_2, \ldots, z_{\ell-1}, z_{\ell-1}, x, y).$$

Since $\mathsf{Spec}(R)$ does not contain even numbers (Case 1), $N(x, y) = \mathrm{NE}(x, y)$. Thus, by Lemma 29 given Cases 2 and 3, $\{R\}$ read-twice-implements EQ_3 or $\{x, \neg x\}$. In the former case, the claim follows. In the latter case, Lemma 26 completes the proof of the claim. ◁

In conclusion, $\mathrm{CSP}_2(R)$ is as hard as $\mathrm{CSP}(R)$ and is thus NP-Hard by Corollary 18. ◀

For symmetric relations of odd arity, we obtain a somewhat weaker result.

▶ **Theorem 33** (Characterization of Odd-Arity Relations). *Let R be a symmetric $(2\ell+1)$-ary relation which is not constantly false. Then* $\mathrm{CSP}_2(R) \in \mathsf{P}$ *in the following cases:*

1. $\mathrm{CSP}_2(R)$ *is trivial.*
2. $\mathsf{Spec}(R)$ *is a zebra.*

Otherwise, $\mathrm{CSP}_2(R)$ *is* NP*-Hard.*

Proof. Case 1 is trivial and Case 2 follows from Corollary 7.. For the NP-Hardness proof, we use a similar argument as in Theorem 31 to conclude that $\mathrm{CSP}_2(R)$ is as hard as $\mathrm{CSP}(R)$. Here is the high-level idea:

- $\{R\}$ read-twice-implements $\mathrm{NE}(x, y)$ or $\{x, \neg x\}$ - Lemma 28.
- $\mathsf{Spec}(R)$ cannot contain two consecutive numbers - Lemma 8.
- $\mathsf{Spec}(R)$ contains a hole of size at least 2 - Observation 2.
- $\{R\}$ read-twice-implements EQ_3 or $\{x, \neg x\}$ - Lemma 30.
- $\{R\}$ read-twice-implements EQ_3 - Lemma 26.

First observe that $\mathsf{Spec}(R)$ cannot contain two consecutive numbers since by Lemma 8, this case is trivial, in the graphical perspective. Consequently, by Observation 2, $\mathsf{Spec}(R)$ must contain a hole of size at least 2. In addition, by Lemma 30, $\{R\}$ read-twice-implements EQ_3 or $\{x, \neg x\}$. In the former case, we are done. In the latter case, Lemma 26 completes the proof. ◀

5 Discussion & Open Questions

In this paper we obtain the first classification of the computational complexity of $\mathrm{CSP}_2(R)$, where R is a single symmetric relation. Alternatively, we obtain a classification of the complexity of the S-factor problem for regular graphs. The characterization is explicit for even degree graphs (even arity), while for odd degrees it states that all nontrivial cases, except for zebras, are NP-hard. An obvious open question is to identify for which sets S, an S-factor is always guaranteed to exist; this amounts to resolving certain open problems in graph theory, even for some small specific S, and looks rather challenging.

More generally, the goal of this line of research is to obtain a complete classification of the computational complexity of $\mathrm{CSP}_2(\Gamma)$, analogous to Schaefer's Dichotomy Theorem. While an explicit classification may encounter difficult graph-theoretic questions, even for some specific Γ, it may well be possible to prove a general complexity dichotomy theorem, as we have done here, without having to resolve explicitly all the hard graph-theoretic questions.

One can observe that all the NP-hardness results of $\mathrm{CSP}_2(\Gamma)$, for the special case when Γ consists of symmetric relation(s), are established via the route of showing that Γ implements the Equality relation. This, in turn, allows to apply Schaefer's Dichotomy Theorem. One interesting open question is whether there exists a set Γ (consisting of not necessarily symmetric relations) that does not implement Equality, yet for which $\mathrm{CSP}_2(\Gamma)$ is NP-Hard. This would imply that Schaefer's Dichotomy does not cover all the cases of bounded read.

References

1 S. Akbari and M. Kano. $\{k, r-k\}$-Factors of r-Regular Graphs. *Graphs and Combinatorics*, 30(4):821–826, 2014. doi:10.1007/s00373-013-1324-x.

2 J. Akiyama and M. Kano. Factors and factorizations of graphs - a survey. *Journal of Graph Theory*, 9(1):1–42, 1985. doi:10.1002/jgt.3190090103.

3 E. Allender, M. Bauland, N. Immerman, H. Schnoor, and H. Vollmer. The complexity of satisfiability problems: Refining Schaefer's theorem. *J. Comput. Syst. Sci.*, 75(4):245–254, 2009. doi:10.1016/j.jcss.2008.11.001.

4 C. Berge. *Graphs and Hypergraphs.* North-Holland mathematical library. Amsterdam, 1973.

5 A. Bernshteyn, O. Khormali, R. R. Martin, J. Rollin, D. Rorabaugh, S. Shan, and A. J. Uzzell. Regular colorings and factors of regular graphs. *CoRR*, abs/1603.09384, 2016. arXiv:1603.09384.

6 A. A. Bulatov. A Dichotomy Theorem for Nonuniform CSPs. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 319–330, 2017. doi:10.1109/FOCS.2017.37.

7 V. Dalmau and D. K. Ford. Generalized Satisfability with Limited Occurrences per Variable: A Study through Delta-Matroid Parity. In *The 28th International Symposium Mathematical Foundations of Computer Science MFCS*, pages 358–367, 2003. doi:10.1007/978-3-540-45138-9_30.

8 Z. Dvorak and M. Kupec. On Planar Boolean CSP. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 432–443, 2015. doi:10.1007/978-3-662-47672-7_35.

9 J. Edmonds. Paths, Trees and Flowers. *CANADIAN JOURNAL OF MATHEMATICS*, pages 449–467, 1965.

10 T. Feder. Fanout limitations on constraint systems. *Theor. Comput. Sci.*, 255(1-2):281–293, 2001. doi:10.1016/S0304-3975(99)00288-1.

11 T. Feder and D. K. Ford. Classification of Bipartite Boolean Constraint Satisfaction through Delta-Matroid Intersection. *SIAM J. Discrete Math.*, 20(2):372–394, 2006. doi:10.1137/S0895480104445009.

12 T. Feder and M. Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. Comput.*, 28(1):57–104, 1998. doi:10.1137/S0097539794266766.

13 T. Gallai. On factorisation of graphs. *Acta Mathematica Academiae Scientiarum Hungarica*, 1(1):133–153, March 1950. doi:10.1007/BF02022560.

14 J. F. Geelen, S. Iwata, and K. Murota. The linear delta-matroid parity problem. *J. Comb. Theory, Ser. B*, 88(2):377–398, 2003. doi:10.1016/S0095-8956(03)00039-X.

15 G. Istrate. Looking for a Version of Schaefer"s Dichotomy Theorem When Each Variable Occurs at Most Twice. Technical report, University of Rochester, Rochester, NY, USA, 1997.

16 M. Kano and H. Matsuda. Partial Parity (g, f)-Factors and Subgraphs Covering Given Vertex Subsets. *Graphs and Combinatorics*, 17(3):501–509, 2001. doi:10.1007/PL00013412.

17 A. Kazda, V. Kolmogorov, and M. Rolínek. Even Delta-Matroids and the Complexity of Planar Boolean CSPs. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 307–326, 2017. doi:10.1137/1.9781611974782.20.

18 L. Lovász. The factorization of graphs. II. *Acta Mathematica Academiae Scientiarum Hungarica*, 23(1):223–246, March 1972. doi:10.1007/BF01889919.

19 H. Lu, D. G. L. Wang, and Q. Yu. On the Existence of General Factors in Regular Graphs. *SIAM J. Discrete Math.*, 27(4):1862–1869, 2013. doi:10.1137/120895792.

20 J. Petersen. Die Theorie der regulären graphs. *Acta Mathematica*, 15:193–220, 1891. doi:10.1007/BF02392606.

21 M. D. Plummer. Graph factors and factorization: 1985–2003: A survey. *Discrete Mathematics*, 307(7-8):791–821, 2007.

22 T. J. Schaefer. The Complexity of Satisfiability Problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC)*, pages 216–226, 1978. doi:10.1145/800133.804350.

23 W.T. Tutte. The Subgraph Problem. In *Advances in Graph Theory*, volume 3 of *Annals of Discrete Mathematics*, pages 289–295. Elsevier, 1978. doi:10.1016/S0167-5060(08)70514-4.

24 D. Zhuk. A Proof of CSP Dichotomy Conjecture. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 331–342, 2017. doi:10.1109/FOCS.2017.38.