# The Complexity of Equivalence and Isomorphism of Systems of Equations over Finite Groups

Gustav Nordh*

Department of Computer and Information Science
Linköpings Universitet
S-581 83 Linköping, Sweden
Fax: +46 13 28 44 99
gusno@ida.liu.se

**Abstract.** We study the computational complexity of the isomorphism and equivalence problems on systems of equations over a fixed finite group. We show that the equivalence problem is in P if the group is Abelian, and coNP-complete if the group is non-Abelian. We prove that if the group is non-Abelian, then the problem of deciding whether two systems of equations over the group are isomorphic is coNP-hard. If the group is Abelian, then the isomorphism problem is graph isomorphism hard. Moreover, if we impose the restriction that all equations are of bounded length, then we prove that the isomorphism problem for systems of equations over finite Abelian groups is graph isomorphism complete. Finally we prove that the problem of counting the number of isomorphisms of systems of equations is no harder than deciding whether there exist any isomorphisms at all.

## 1 Introduction

The computational complexity of deciding whether systems of equations over a fixed finite group are solvable has been studied in the past. In [5], it is proved that the problem is in P if the group is Abelian and NP-complete otherwise. This line of research continued in [8, 10], where the corresponding problem for finite monoids was given a complete solution. Some partial results in the general case of finite semigroups have been proved in [10]. Note that even the restricted problem of determining the computational complexity of solving systems of equations over a fixed regular semigroup is still open. The problem of deciding whether systems of equations over a fixed finite group $(G, \cdot)$ are solvable is denoted $\textsc{Eqn}^*_G$ in the literature.

The computational complexity of counting solutions to systems of equations over a fixed finite semigroup has been studied in [9], where it is proved that if the semigroup is an Abelian group, then the problem is in FP, and if the semigroup is a non-Abelian group, then the problem is #P-complete. This problem is denoted $\#\textsc{Eqn}^*_G$.

* Supported by the *National Graduate School in Computer Science* (CUGS), Sweden.

In this paper we study the computational complexity of deciding whether systems of equations over a fixed finite group are equivalent/isomorphic. We also study the problem of counting the number of isomorphisms. These fundamental problems have as far as we know eluded previous investigations from a computational perspective, except for some results on the Boolean constraint equivalence and isomorphism problems [2, 3] that are also relevant in our setting. More specifically the equivalence problem for systems of equations of bounded length over the two element group $\mathbb{Z}_2$ is in P, and the corresponding isomorphism problem is GRAPH ISOMORPHISM-complete. The computational complexity of several other isomorphism and equivalence problems have been intensively studied in the past, most notably the graph isomorphism problem [6] and the formula isomorphism problem [1].

A system of equations over a fixed finite group $(G, \cdot)$ is a collection of equations of the form $x_1 \cdot x_2 \cdot \ldots \cdot x_k = x_{k+1} \cdot \ldots \cdot x_n$, where each $x_i$ is either a variable or a constant in $G$.

**Definition 1.** *Let $S$ be a system of equations on variables $X$ and let $\pi$ be a permutation of $X$. By $\pi(S)$ we denote the system of equations that results when we replace all variables $x$ in $S$ by $\pi(x)$.*

- EQUIV-EQN$^*_G$ *is the problem of deciding whether two systems of equations $S_1$ and $S_2$ on variables $X$ over $(G, \cdot)$ are equivalent, i.e., whether for every assignment of the variables in $X$ to values in $G$, $S_1$ is satisfied if and only if $S_2$ is satisfied. Note that when we say that $S_1$ and $S_2$ are systems of equations on variables $X$ we only mean that $X$ is the union of the variables in $S_1$ and $S_2$, hence all variables in $X$ need not occur both in $S_1$ and $S_2$. If $S_1$ is equivalent to $S_2$ we denote this by $S_1 \equiv S_2$.*
- ISO-EQN$^*_G$ *is the problem of deciding whether two systems of equations $S_1$ and $S_2$ on variables $X$ over $(G, \cdot)$ are isomorphic, i.e., whether there exist a permutation $\pi$ of the variables in $X$ such that $\pi(S_1) \equiv S_2$. If $S_1$ is isomorphic to $S_2$ we denote this by $S_1 \cong S_2$.*
- #ISO-EQN$^*_G$ *is the counting version of* ISO-EQN$^*_G$*, i.e., the problem of counting the number of permutations $\pi$ of the variables in $X$ such that $\pi(S_1) \equiv S_2$.*

The complexity of EQUIV-EQN$^*_G$ (ISO-EQN$^*_G$, #ISO-EQN$^*_G$) is measured in the size of the systems of equations (the size of $(G, \cdot)$ is fixed and does not matter).

We prove that if the group is non-Abelian the equivalence problem is coNP-complete and if the group is Abelian the equivalence problem is in P. When it comes to the isomorphism problem we prove that if the group is non-Abelian, the isomorphism problem is coNP-hard. If the group is Abelian, then the isomorphism problem is GRAPH ISOMORPHISM-hard. If we impose the restriction that all equations are of bounded length, then the isomorphism problem for systems of equations over finite Abelian groups becomes GRAPH ISOMORPHISM-complete. For the problem of counting the number of isomorphisms, we give an algorithm that shows that it is no harder to count the number of isomorphisms than to decide whether any isomorphisms exists at all. As a corollary to this algorithm we obtain the result that $\textsc{Iso-Eqn}^*_G$ for Abelian groups is powerless as an oracle to PP, and $\textsc{Iso-Eqn}^*_G$ for non-Abelian groups is no more powerful than an NP-oracle for PP. These results indicates that $\textsc{Iso-Eqn}^*_G$ for Abelian groups is not NP-complete, and that $\textsc{Iso-Eqn}^*_G$ for non-Abelian groups is not $\Sigma_2^{\mathrm{P}}$-complete.

The paper is organized as follows. In Sect. 2 we prove our results concerning the equivalence problem, Sect. 3 deals with the isomorphism problem, Sect. 4 treats the corresponding counting problem, and finally in Sect. 5 we present our conclusions and some ideas for future research.

## 2 Equivalence

In this section we investigate the computational complexity of $\textsc{Equiv-Eqn}^*_G$, that is, the problem of deciding whether two systems of equations over a fixed finite group are equivalent. We prove that $\textsc{Equiv-Eqn}^*_G$ is coNP-complete if $(G, \cdot)$ is non-Abelian, and that $\textsc{Equiv-Eqn}^*_G$ is in P if $(G, \cdot)$ is an Abelian group.

First note that it is easy to see that the equivalence problem is in coNP.

**Theorem 1.** $\textsc{Equiv-Eqn}^*_G$ *is in coNP.*

*Proof.* Nondeterministically choose an assignment of the variables in $S_1$ and $S_2$ to values in $G$. It can be checked in polynomial time whether this assignment satisfies exactly one of $S_1$ and $S_2$. Hence the problem of deciding whether two systems of equations $S_1$ and $S_2$ over $(G, \cdot)$ are inequivalent, is in NP. Thus it follows that the problem of deciding whether two systems of equations over a fixed finite group are equivalent is in coNP. □

The following theorem states that if it is hard to decide whether systems of equations over a group $(G, \cdot)$ are solvable, then it is also hard to decide whether systems of equations over the same group are equivalent.

**Theorem 2.** *If* $\textsc{Eqn}^*_G$ *is NP-complete then* $\textsc{Equiv-Eqn}^*_G$ *is coNP-complete.*

*Proof.* If $\textsc{Eqn}^*_G$ is NP-complete it follows that it is coNP-complete to decide whether a system of equations over $(G, \cdot)$ is insoluble. Since a system of equations is insoluble if and only if it is equivalent to an insoluble system of equations (for example a system of equations containing the equation $0 = 1$), it follows that $\textsc{Equiv-Eqn}^*_G$ is coNP-complete. □

The previous theorem and the fact that $\textsc{Eqn}^*_G$ is NP-complete when $(G, \cdot)$ is a non-Abelian group [5] immediately implies the following corollary.

**Corollary 1.** $\textsc{Equiv-Eqn}^*_G$ *is coNP-complete when* $(G, \cdot)$ *is a non-Abelian group.*

Next we prove that if it is easy to count the number of solutions to systems of equations over a group $(G, \cdot)$, then it is also easy to decide whether systems of equations over the same group are equivalent.

**Theorem 3.** Equiv-Eqn$^*_G$ *is in P if* #Eqn$^*_G$ *is in FP.*

*Proof.* Let $S_1$ and $S_2$ be two systems of equations. Count the number of solutions to $S_1$ and $S_2$, if they have different number of solutions they are not equivalent. Thus we can assume that $S_1$ and $S_2$ have the same number of solutions. Count the number of solutions to the system of equations consisting of the union of $S_1$ and $S_2$. If the number of solutions to this system of equations equals the number of solutions to $S_1$, we know that $S_1$ and $S_2$ have the same set of solutions and hence $S_1$ is equivalent to $S_2$, otherwise $S_1$ and $S_2$ are inequivalent. □

The following corollary follows directly from Theorem 3 and the fact that #Eqn$^*_G$ is in FP when $(G, \cdot)$ is an Abelian group [9].

**Corollary 2.** Equiv-Eqn$^*_G$ *is in P when* $(G, \cdot)$ *is an Abelian group.*

It should be clear that Theorem 2 and Theorem 3 also holds when generalized to systems of equations over a fixed finite semigroup $(S, \cdot)$. Hence interesting results on the complexity of Equiv-Eqn$^*_S$, where $(S, \cdot)$ is finite semigroup, can be deduced from the results for #Eqn$^*_S$ and Eqn$^*_S$ proved in [8–10]. We collect these results in the following corollary.

**Corollary 3.** Equiv-Eqn$^*_S$ *is coNP-complete when*

- $(S, \cdot)$ *is a monoid but is not in the join of Abelian groups and idempotent Abelian monoids [8],*
- $(S, \cdot)$ *is a band but is not a normal band [10],*
- $(S, \cdot)$ *is a regular semigroup but is not a strong normal band of Abelian groups [10].*

Equiv-Eqn$^*_S$ *is in P when*

- $(S, \cdot)$ *is a direct product of an Abelian group and a rectangular band [9],*
- $(S, \cdot)$ *is a semigroup with zero, such that for all elements* $x, y$, $x \cdot y = 0$ *[9].*

## 3 Isomorphism

In this section we investigate the computational complexity of Iso-Eqn$^*_G$, that is, the problem of deciding whether two systems of equations over a fixed finite group are isomorphic. We prove that Iso-Eqn$^*_G$ is coNP-hard if $(G, \cdot)$ is non-Abelian, and that Iso-Eqn$^*_G$ is Graph Isomorphism-hard if $(G, \cdot)$ is an Abelian group. If we restrict the problem and demand that all equations are of bounded length, then Iso-Eqn$^*_G$ in the Abelian case becomes Graph Isomorphism-complete, and Iso-Eqn$^*_G$ in the non-Abelian case is in $\mathrm{P}^{\mathrm{NP}}_{||}$, i.e., the class of problems solvable in polynomial time with parallel access to an NP-oracle.

We begin by giving upper bounds for the complexity of the isomorphism problem.

**Theorem 4.** Iso-Eqn$^*_G$ *is in NP when* $(G, \cdot)$ *is an Abelian group and* Iso-Eqn$^*_G$ *is in* $\Sigma^{\mathrm{P}}_2$ *when* $(G, \cdot)$ *is a non-Abelian group.*

*Proof.* We begin by proving the $\Sigma_2^{\mathrm{P}}$ upper bound. Nondeterministically choose a permutation $\pi$ of the variables in $X$ and use an NP-oracle to check whether $\pi(S_1) \equiv S_2$. Hence ISO-EQN$^*_G$ is in $\mathrm{NP}^{\mathrm{NP}} = \Sigma_2^{\mathrm{P}}$.

The NP upper bound for ISO-EQN$^*_G$ when $(G, \cdot)$ is an Abelian group follows from the results in the previous section on the equivalence problem. We know from Corollary 2 that $\pi(S_1) \equiv S_2$ can be decided in polynomial time when $(G, \cdot)$ is an Abelian group, hence ISO-EQN$^*_G$ is in NP when $(G, \cdot)$ is an Abelian group. □

The following theorem states that if it is hard to decide whether systems of equations over a group $(G, \cdot)$ are solvable, then it is also hard to decide whether systems of equations over the same group are isomorphic.

**Theorem 5.** *If* EQN$^*_G$ *is NP-complete then* ISO-EQN$^*_G$ *is coNP-hard.*

*Proof.* If EQN$^*_G$ is NP-complete, then it follows that it is coNP-hard to decide whether a system of equations over $(G, \cdot)$ is insoluble. Since a system of equations is insoluble if and only if it is isomorphic to an insoluble system of equations, e.g., a system of equations containing the equation $0 = 1$, it follows that ISO-EQN$^*_G$ is coNP-hard. □

The previous theorem and the fact that EQN$^*_G$ is NP-complete when $(G, \cdot)$ is a non-Abelian group [5] immediately implies the following corollary.

**Corollary 4.** ISO-EQN$^*_G$ *is coNP-hard when* $(G, \cdot)$ *is a non-Abelian group.*

The following theorem indicates that ISO-EQN$^*_G$ for Abelian groups perhaps is not in P, or at least that it is hard to prove that ISO-EQN$^*_G$ is in P for Abelian groups.

**Theorem 6.** *Let* $(G, \cdot)$ *be a finite Abelian group, then* GRAPH ISOMORPHISM *is polynomial-time many-one reducible to* ISO-EQN$^*_G$.

*Proof.* It is known that GRAPH ISOMORPHISM is polynomial-time equivalent to the graph isomorphism problem restricted to bipartite graphs [4]. To simplify the proof we assume from now on that all graphs are bipartite and contains no isolated vertices. Let $T$ be the following transformation from a graph $H = (V, E)$, where $V = \{1, 2, \ldots, n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$, to a system of equations $S$ over $(G, +)$:

$$T(H) = \{x_i + x_j + y_k = 1 \mid e_k = \{i, j\}\} \cup \{x_i + z_i + z'_i = 1 \mid i \in V\}.$$

Note that since the group is Abelian we adopt the common notation and denote the group operation by $+$. Furthermore, as we will explain later, it is sufficient to consider the case where $(G, +)$ is a cyclic group (that is groups of the form $\mathbb{Z}_{p^r}$ for some prime $p$ and positive integer $r$). We let 1 denote one of the group generators, and 0 the identity element.

The proof of the theorem relies on the following lemma, which shows that $T(H)$ is a maximum set of equations of type, $a + b + c = 1$. This property is important since, checking whether two maximum sets of equations are equivalent basically amounts to checking whether the sets are equal. This is the same approach as previously used in the treatment of the isomorphism problem for Boolean constraints in [2]. Note that $T(H)$ would not necessary be a maximum set of equations of type $a+b+c=1$ for non-bipartite graphs $H$.

Let $T(H) \rightarrow a+b+c=1$ (where $a, b, c$ are variables occurring in $T(H)$) denote that the equation $a+b+c=1$ can be inferred from the system of equations, $T(H)$, i.e., there exist no assignment of the variables in $T(H)$ such that $T(H)$ is satisfied and $a+b+c=1$ is not satisfied. Note that since $(G,+)$ is Abelian, we (for example) consider $a+b+c=1$ as being identical to $b+a+c=1$.

**Lemma 1.** *The following holds for every triple $a, b, c$ of distinct variables in $T(H)$: If $T(H) \rightarrow a+b+c=1$, then $a+b+c=1 \in T(H)$.*

The proof of the lemma is a straightforward but tedious case analysis that we have to omit due to space limitations. The importance of the fact that $T(H)$ is a maximum set of equations of the form $a+b+c=1$ lies in the observation that $T(H_1) \equiv T(H_2)$ if and only if $T(H_1) = T(H_2)$. Note again that since $(G,+)$ is Abelian we consider $a+b+c=1$ as being identical to $b+c+a=1$, and so on. Hence $T(H_1)$ is isomorphic to $T(H_2)$ if and only if there exists a permutation $\rho$ of the variables in $T(H_1) \cup T(H_2)$ such that $\rho(T(H_1)) = T(H_2)$.

Let $H_1 = (V, E)$ and $H_2 = (V, E')$ be two bipartite graphs, with an equal number of vertices and edges, where $V = \{1, 2, \ldots, n\}$, $E = \{e_1, e_2, \ldots, e_m\}$, and $E' = \{e'_1, e'_2, \ldots, e'_m\}$. If $\pi : V \rightarrow V$ is an isomorphism from $H_1$ to $H_2$ then $\rho$ is an isomorphism from $T(H_1)$ to $T(H_2)$ defined as follows:

- $\rho(x_i) = x_{\pi(i)}$, $\rho(z_i) = z_{\pi(i)}$, $\rho(z'_i) = z'_{\pi(i)}$, for $i \in V$;
- For $e_k = \{i, j\}$, $\rho(y_k) = y_l$ where $e'_l = \{\pi(i), \pi(j)\}$.

For the converse, suppose that $\rho$ is an isomorphism from $T(H_1)$ to $T(H_2)$. As we have already observed $\rho(T(H_1)) = T(H_2)$. Now consider the properties of the different classes of variables.

- Variables from $X$ are exactly those variables that occur at least twice and that also occur in an element of $T(H)$ together with two variables that occur exactly once. So, $\rho$ will map $X$ onto $X$.
- Elements of $Z$ are those variables that occur exactly once and that occur together with an element from $X$ and another element that occurs exactly once. So, $\rho$ will map $Z$ to $Z$.
- Everything else is an element of $Y$. So, $\rho$ will map $Y$ to $Y$.

For $i \in V$, define $\pi(i) = j$ if and only if $\rho(x_i) = x_j$. $\pi$ is one-to-one and onto by the observation above. What remains to be proved is that $\{i, j\} \in E$ if and only if $\{\pi(i), \pi(j)\} \in E'$. Let $e_k = \{i, j\}$. Then $x_i + x_j + e_k = 1 \in T(H_1)$. Hence, $\rho(x_i) + \rho(x_j) + \rho(y_k) = 1 \in T(H_2)$, i.e., $x_{\pi(i)} + x_{\pi(j)} + \rho(y_k) = 1 \in T(H_2)$. This implies that $\rho(y_k) = y_l$ where $e'_l = \{\pi(i), \pi(j)\}$, hence $\{\pi(i), \pi(j)\} \in E'$. For the converse, suppose that $\{\pi(i), \pi(j)\} \in E'$. Then $x_{\pi(i)} + x_{\pi(j)} + y_l = 1 \in T(H_2)$ for $e_l = \{\pi(i), \pi(j)\}$, and thus $x_i + x_j + \rho^{-1}(y_l) \in T(H_1)$. By the structure of $T(H_1)$, it follows that $\{i, j\} \in E$. Hence we have proved that Graph Isomorphism is polynomial-time many-one reducible to Iso-Eqn$^*_G$ where $(G, +)$ is a finite cyclic group.

The fundamental theorem of finitely generated Abelian groups makes it possible to extended this result to all finite Abelian groups. Note that all finite Abelian groups are of course finitely generated.

**Lemma 2.** *Every finite Abelian group $(G, +)$ is isomorphic to a direct product of cyclic groups in the form $\mathbb{Z}_{p_1^{r_1}} \times \mathbb{Z}_{p_2^{r_2}} \times \ldots \times \mathbb{Z}_{p_n^{r_n}}$, where the $p_i$'s are (not necessarily distinct) primes, and $|G| = p_1^{r_1} \cdot \ldots \cdot p_n^{r_n}$.*

Thus given a system of equations $S_1$ over a finite Abelian group $(G, +)$, we can use the fundamental theorem of finitely generated Abelian groups to view it as $n$ independent systems of equations, each over a cyclic group $\mathbb{Z}_{p_i^{r_i}}$. It should be clear that two systems of equations $S_1$ and $S_2$ over $(G, +)$ are isomorphic if and only if $S_1$ is isomorphic to $S_2$ when viewed as systems of equations over $\mathbb{Z}_{p_i^{r_i}}$, for all $i \in \{1, \ldots, n\}$. We have already proved that $H_1$ is isomorphic to $H_2$ if and only if $T(H_1)$ is isomorphic to $T(H_2)$, when $T(H_1)$ and $T(H_2)$ are systems of equations over an arbitrary cyclic group. Hence given a finite Abelian group $(G, +)$ and two graphs $H_1$ and $H_2$, then $H_1$ is isomorphic to $H_2$ if and only if the two systems of equations $T(H_1)$ and $T(H_2)$ over $(G, +)$ are isomorphic. $\square$

Despite intensive research Graph Isomorphism is not known to be in coNP. Hence in light of the previous theorem it seems hard to prove that Iso-Eqn$^*_G$ is in coNP when $(G, \cdot)$ is an Abelian group.

**Corollary 5.** Iso-Eqn$^*_G$ *for Abelian groups is not in coNP unless* Graph Isomorphism *is in coNP.*

Let Iso-B-Eqn$^*_G$ denote the problem of deciding whether two systems of equations over $(G, \cdot)$, where each equation have bounded length, i.e., every equation are of length at most $k$ for some constant $k$, are isomorphic.

**Theorem 7.** Iso-B-Eqn$^*_G$ *is* Graph Isomorphism*-complete when $(G, \cdot)$ is an Abelian group.*

*Proof.* By Theorem 6 we know that Graph Isomorphism is polynomial-time reducible to Iso-B-Eqn$^*_G$. Hence what remains is to give a polynomial-time reduction from Iso-B-Eqn$^*_G$ to Graph Isomorphism. We will actually give a reduction from Iso-B-Eqn$^*_G$ to the Graph Isomorphism-complete problem, vertex colored graph isomorphism (VCGI) [6].

**Definition 2.** *VCGI is the problem of, given two vertex-colored graphs $H_1 = (V_1, E_1, \chi_1)$, $H_2 = (V_2, E_2, \chi_2)$, $\chi_1, \chi_2 \colon V \to N$, to determine whether there exist an isomorphism from $H_1$ to $H_2$ that preserves colors, i.e., whether there exists a bijection $\pi : V_1 \to V_2$ such that $\{v, v'\} \in E_1$ if and only if $\{\pi(v), \pi(v')\} \in E_2$, and $\chi_1(v) = \chi_2(\pi(v))$ for all $v \in V_1$.*

Let $S_1$ and $S_2$ be two systems of equations of bounded length over an Abelian group $(G, \cdot)$ on variables $X$. We will first bring $S_1$ and $S_2$ into normal form. We choose an approach similar to that in Theorem 6 and reduce $S_1$ and $S_2$ to the maximum set of equations of length at most $k$ that can be inferred from $S_1$ and $S_2$ respectively.

Let $\langle S_1 \rangle$ denote the set of all equations $E$ of length less than $k$ such that all of $E$'s variables occur in $X$ and $S_1 \to E$. It should be clear that $S_1$ is equivalent to $\langle S_1 \rangle$, since $S_1 \subseteq \langle S_1 \rangle$ and $S_1 \to \langle S_1 \rangle$. $\langle S_2 \rangle$ is defined analogously. Note that $\langle S_1 \rangle$ and $\langle S_2 \rangle$ can be computed in polynomial time (in $|S_1| + |S_2|$), since there exist at most $O(|X|^k)$ equations $E$ of length less than $k$ such that all of $E$'s variables occur in $X$, and since $(G, \cdot)$ is Abelian we can use the polynomial time algorithm for $\text{EQN}^*_G$ to decide whether $S \to E$. Note that $S \to E$ if and only if for every assignment to the variables in $E$ such that $E$ is not satisfied, this assignment applied to $S$ makes $S$ insoluble. Since there is at most $O(|G|^k)$ possible assignments of the variables in $E$, and testing whether such an assignment makes $S$ insoluble can be done in polynomial time (by the result in [5]), it follows that $S \to E$ can be decided in polynomial time. It should be clear that if $\langle S_1 \rangle$ is equivalent to $\langle S_2 \rangle$ then $\langle S_1 \rangle = \langle S_2 \rangle$, hence if $\pi$ is a permutation of the variables in $X$ such that $\pi(\langle S_1 \rangle) \equiv \langle S_2 \rangle$, then $\pi(\langle S_1 \rangle) = \langle S_2 \rangle$.

Now we proceed and encode $\langle S_1 \rangle$ and $\langle S_2 \rangle$ as vertex colored graphs $H_1$ and $H_2$ such that there exist a permutation $\pi$ of $X$ such that $\pi(\langle S_1 \rangle) \equiv \langle S_2 \rangle$ if and only if $(H_1, H_2) \in VCGI$.

Let $\langle S_1 \rangle$ consist of the equations $A_1, \ldots, A_m$, that is, $\langle S_1 \rangle =$

$$\begin{array}{rcll}
x_{11} + x_{12} + \ldots + x_{1j_1} & = & x_{1j_1+1} + \ldots + x_{1k_1} & (A_1) \\
x_{21} + x_{22} + \ldots + x_{2j_2} & = & x_{2j_2+1} + \ldots + x_{2k_2} & (A_2) \\
& \vdots & & \\
x_{m1} + x_{m2} + \ldots + x_{mj_m} & = & x_{mj_m+1} + \ldots + x_{mk_m} & (A_m)
\end{array}$$

Let $T(\langle S_1 \rangle) = H_1 = (V, E, \chi)$ be the following vertex colored graph:

- $V = \{0, 1, \ldots, |G| - 1\} \cup \{x \mid x \in X\} \cup \{a_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq k_i\} \cup \{A_i \mid 1 \leq i \leq m\}$. Hence, the set of vertices corresponds to the elements in $(G, \cdot)$, the variables in $X$, the elements in the equations in $\langle S_1 \rangle$, and the equations in $\langle S_1 \rangle$.
- $E = \{\{x, a_{ij}\} \mid x = a_{ij}\} \cup \{\{a_{ij}, A_i\} \mid 1 \leq i \leq m, 1 \leq j \leq k_i\}$.
- The vertex coloring $\chi$ is used to distinguish the different categories of vertices. Of course, we must allow any permutation of the variables, so all vertices corresponding to variables in $X$ are assigned the same color. We also need to allow the permutation of vertices corresponding to equations, and because the group is Abelian we allow the permutation of vertices corresponding to elements within the same side of each equation.
  - $\chi(i) = i$,
  - $\chi(x) = |G|$ for all $x \in X$,
  - $\chi(A_r) = |G| + 1$,
  - $\chi(a_{ij}) = |G| + 2$ if $j \leq j_i$,

- $\chi(a_{ij}) = |G| + 3$ if $j > j_i$

Define $T(\langle S_2 \rangle) = H_2$ in the analogous way.

If there is a permutation $\pi$ of $X$ such that $\pi(\langle S_1 \rangle) = \langle S_2 \rangle$, then it is easy to see that $(T(\langle S_1 \rangle), T(\langle S_2 \rangle)) \in VCGI$. On the other hand, if $(T(\langle S_1 \rangle), T(\langle S_2 \rangle)) \in VCGI$ by a permutation $\pi$ on $T(\langle S_1 \rangle)$ it should be clear that $\pi(\langle S_1 \rangle) = \langle S_2 \rangle$. Just note that any permutation of vertices corresponding to equations forces a permutation of the vertices corresponding to elements in the equations. Thus if $\pi(A_i) = A_j$ then necessarily $\pi(a_{it}) = a_{js}$, for all $1 \le t \le k_i$ and some $1 \le s \le k_j$. The only remaining part of the permutation in $T(\langle S_1 \rangle)$ is the permutation of vertices corresponding to variables in $X$, hence $\pi(\langle S_1 \rangle) = \langle S_2 \rangle$. □

Using the same idea as in the proof of the preceding theorem we can prove a tighter upper bound for Iso-B-Eqn$^*_G$, compared to the trivial $\Sigma_2^{\mathrm{P}}$ upper bound from Theorem 4. More specifically we prove that Iso-B-Eqn$^*_G$ is in $\mathrm{P}_{||}^{\mathrm{NP}}$, the class of problems solvable in polynomial time with parallel (i.e., truth-table) access to an NP-oracle.

**Theorem 8.** Iso-B-Eqn$^*_G$ *is in* $P_{||}^{NP}$.

*Proof.* Let $S_1$ and $S_2$ be two systems of equations on variables $X$ over a non-Abelian group $(G, \cdot)$. We use the same normal form as in the proof of the preceding theorem, i.e., we compute $\langle S_1 \rangle$ and $\langle S_2 \rangle$. Following the proof of the preceding theorem it is easy to verify that $\langle S_1 \rangle$ and $\langle S_2 \rangle$ can be computed in polynomial time with parallel access to an NP-oracle. Now the existence of a permutation $\pi$ of the variables in $X$ such that $\pi(\langle S_1 \rangle) = \langle S_2 \rangle$ can be determined with one query to an NP-oracle. Hence, it can be decided in polynomial time with two rounds of parallel queries to NP whether $S_1$ and $S_2$ are isomorphic, and thus Iso-B-Eqn$^*_G$ is in $\mathrm{P}_{||}^{\mathrm{NP}}$. □

## 4 Counting Isomorphisms

Mathon showed that the counting version of Graph Isomorphism is polynomial time Turing reducible to the decision version [7]. Thus Graph Isomorphism behaves differently than the known NP-complete problems. This was historically the first hint that Graph Isomorphism might not be NP-complete. We prove analogous results for #Iso-Eqn$^*_G$.

Given a system of equations over a finite group $(G, \cdot)$ we are interested in the set of permutations $\pi$ of the variables in $X$ such that $\pi(S) \equiv S$. It is easy to see that this set of permutations forms a group, denoted $aut(S)$, the automorphism group of $S$. Aut-Eqn$^*_G$ is the problem of computing the automorphism group of a system of equations $S$ over $(G, \cdot)$. #Aut-Eqn$^*_G$ is the corresponding counting problem.

The following theorem states that it is no harder to count the number of isomorphisms between two systems of equations over a fixed finite group than to decide whether an isomorphism exists at all. This indicates that Iso-Eqn$^*_G$ is not NP-complete for Abelian groups, and that Iso-Eqn$^*_G$ is not $\Sigma_2^{\mathrm{P}}$-complete for non-Abelian groups.

**Theorem 9.** *$\#$*Iso-Eqn$^*_G$ *is Turing equivalent to* Iso-Eqn$^*_G$.

*Proof.* The proof is based on the same principle as the analogous proof for Graph Isomorphism due to Mathon [7]. First note that Iso-Eqn$^*_G$ is trivially polynomial time reducible to $\#$Iso-Eqn$^*_G$. Let $N_1$ be the number of permutations $\pi$ of $X$ such that $\pi(S_1) \equiv S_2$. If $S_1 \not\cong S_2$, i.e., there exist no permutation $\pi$ of $X$ such that $\pi(S_1) \equiv S_2$, then $N_1 = 0$. Otherwise we claim that $N_1 = |aut(S_1)| = |aut(S_2)|$. This follows easily from the fact that if $\pi$ is a permutation such that $\pi(S_1) \equiv S_2$, and $\rho \in aut(S_1)$ then $\rho \circ \pi$ is also a permutation such that $\rho \circ \pi(S_1) \equiv S_2$ (note that $\rho \circ \pi$ denotes the composition where $\rho$ is applied prior to $\pi$, hence $\rho \circ \pi(S_1) = \pi(\rho(S_1))$). Moreover any permutation $\pi'$ such that $\pi'(S_1) \equiv S_2$ can be uniquely expressed in the form $\pi' = \rho' \circ \pi$, where $\rho' \in aut(S_1)$. Thus the set of isomorphisms from $S_1$ to $S_2$ is the right coset, $aut(S_1) \circ \pi$, where $\pi$ is an isomorphism from $S_1$ to $S_2$. Hence $\#$Iso-Eqn$^*_G$ is Turing reducible to Iso-Eqn$^*_G$ if $\#$Aut-Eqn$^*_G$ is Turing reducible to Iso-Eqn$^*_G$, which we prove next. ☐

**Lemma 3.** *$\#$*Aut-Eqn$^*_G$ *is Turing reducible to* Iso-Eqn$^*_G$.

*Proof.* First note that we can determine whether a system of equations $S$ is solvable or not by making a single query to Iso-Eqn$^*_G$. If $S$ is insoluble then of course $|aut(S)| = |X|!$.

Given a solvable system of equations $S$ we need a construction that essentially forces a variable to be mapped to itself under any permutation $\rho \in aut(S)$. We call variables $x_i$ and $x_j$ equivalent with respect to $S$ if, for any assignment $\alpha$ that satisfies $S$, we have $\alpha(x_i) = \alpha(x_j)$. By $E_S(x_i)$ we denote the set of variables of $S$ that are equivalent to $x_i$. Consider any automorphism $\rho \in aut(S)$, if $\rho$ maps $x_i$ to $x_k$, then clearly $\rho(E_S(x_i)) = E_S(x_k)$.

This property can be used to force a variable to be mapped to itself (or an equivalent variable), since $\rho$ always maps $x_i$ to a variable $x_k$ such that $|E_S(x_i)| = |E_S(x_k)|$. The trick is to make $|E_S(x_i)|$ unique by introducing new equations of the form $x_i = z_j$, where $z_j$ is a new variable called a labelling variable. Let $L(x_i, t)$ denote $\bigcup_{j=1}^{t} x_i = z_j$, and $S_{[I]}$ the system of equations where all (inequivalent) variables in $I$ have been given a unique label.

We construct the system of equations $S_{[I]}$ by successively labelling the variables in $I$. Let $S_0 = S$. Suppose that we have already labelled the first $k$ variables in $I$ and obtained the system of equations $S_k$ for some $k \geq 0$. Let $x_i$ be the next variable in $I$ to be labelled. That is, $x_i$ is not equivalent to any of the variables already labelled. Let $t$ be the least integer such that $|E_{S_k}(x_i)| + t \neq |E_{S_k}(x_j)|$ for any $x_j \notin E_{S_k}(x_i)$, then let $S_{k+1} = S_k \cup L(x_i, t)$. Note that when $S$ is a system of equations over an Abelian group, then $|E_S(x_i)|$ can be computed in polynomial time, just check whether $S \equiv S \cup x_i = x_j$ for all $x_j$ in $X$ (remember that by Corollary 2 we know that equivalence of systems of equations over Abelian groups is in P). In the case where $S$ is a system of equations over a non-Abelian group we can use an NP-oracle to check whether $S \equiv S \cup x_i = x_j$ for all $x_j$ in $X$. Since Iso-Eqn$^*_G$ for non-Abelian groups is coNP-hard, we can in particular use this method when we have Iso-Eqn$^*_G$ as an oracle.

The procedure above is repeated at most $|I| \leq n$ times and the number $t$ is bounded by $n$ in each iteration, where $n = |X|$. Hence $S_{[I]}$ can be constructed in polynomial time in the Abelian case, and in polynomial time with Iso-Eqn$^*_G$ as an oracle in the non-Abelian case.

Now with the technicalities of the labelling out of the way we can proceed to compute $|aut(S)|$. Let $S_{[x_1,\ldots,x_n]}$ be a system of equations where all variables have been labelled in the way described above. First note that we are only interested in automorphisms that acts as the identity mapping on the variables introduced in the labelling and the variables still labelled, this subgroup of the full automorphism group is denoted $aut'(S)$. Hence $|aut'(S_{[x_1,\ldots,x_n]})| = 1$. The crucial observation is that $|aut'(S_{[x_1,\ldots,x_{i-1}]})| = d_i|aut'(S_{[x_1,\ldots,x_{i-1},x_i]})|$, where $d_i$ is the size of the orbit of $x_i$ under $aut'(S_{[x_1,\ldots,x_{i-1}]})$. It should be clear that $|aut'(S_0)| = |aut(S)| = d_1 d_2 \ldots d_n$. The fact that the order of a permutation group can be computed from the size of the orbits in the way described above is a standard result from group theory, consult [6] for the details.

The orbit of $x_i$ under $aut'(S_{[x_1,\ldots,x_{i-1}]})$ can be found by making $n-i$ queries to Iso-Eqn$^*_G$. Ask the query $S_{[x_1,\ldots,x_{i-1}]}[x_i] \cong S_{[x_1,\ldots,x_{i-1}]}[x_j]$ for each variable $x_j$, $j \geq i$, where $S_{[x_1,\ldots,x_{i-1}]}[x_i]$ and $S_{[x_1,\ldots,x_{i-1}]}[x_j]$ denotes systems of equations where $x_i$ and $x_j$ have been assigned the same (unique) label (using the method described above). If the answer is yes, we know that $x_j$ is in the orbit of $x_i$ under $aut'(S_{[x_1,\ldots,x_{i-1}]})$. Hence $|aut(S)| = d_1 d_2 \ldots d_n$ can be computed by making $O(n^2)$ queries to Iso-Eqn$^*_G$ and thus #Aut-Eqn$^*_G$ is Turing reducible to Iso-Eqn$^*_G$. □

Mathon's algorithm for computing the number of isomorphisms of two graphs, have the interesting property that at each intermediate stage the number of isomorphisms of the labelled graphs are known. This property has been exploited to prove that Graph Isomorphism (GI) is low for PP, i.e., Graph Isomorphism is powerless as an oracle to PP ($\mathrm{PP}^{\mathrm{GI}} = \mathrm{PP}$) [6]. This is generally interpreted as further evidence for the hypothesis that Graph Isomorphism is not NP-complete. For the details and significance of lowness results, again consult [6]. The properties of our algorithm (that is based on Mathon's algorithm) for computing #Iso-Eqn$^*_G$ together with the same line of reason as that preceding Theorem 5.3 in [1] implies the following lowness results.

**Corollary 6.** *$PP^{IAb} = PP$, and $PP^{IG} = PP^{NP}$, where IAb denotes* Iso-Eqn$^*_G$ *for Abelian groups and IG denotes* Iso-Eqn$^*_G$ *for non-Abelian groups.*

Hence Iso-Eqn$^*_G$ for Abelian groups is powerless as an oracle to PP, and Iso-Eqn$^*_G$ for non-Abelian groups is no more powerful than an NP-oracle for PP.

## 5 Conclusions

We give a dichotomy for the complexity of Equiv-Eqn$^*_G$ for all finite groups $(G, \cdot)$. A natural direction for future research would be to prove a similar dichotomy for Equiv-Eqn$^*_S$ for all finite semigroups $(S, \cdot)$.

When it comes to the complexity of ISO-EQN$^*_G$, the situation is not as clear. We prove the upper bound of $\Sigma_2^{\mathrm{P}}$ and the lower bound of coNP-hard in the non-Abelian case, and the upper bound of NP and the lower bound of GRAPH ISOMORPHISM-hard in the Abelian case. But the results in Theorem 9 and Corollary 6 give strong indications that these upper bounds ($\Sigma_2^{\mathrm{P}}$ and NP respectively) are not tight.

Moreover, it has been proved, by the use of interactive proofs, that GRAPH ISOMORPHISM and formula isomorphism is not NP-complete and $\Sigma_2^{\mathrm{P}}$-complete respectively, unless the polynomial hierarchy collapses [1, 6]. It would be interesting to investigate whether similar techniques can be used to prove analogous results for ISO-EQN$^*_G$. Another question left open by the present paper is the existence of a polynomial time reduction from ISO-EQN$^*_G$ to ISO-B-EQN$^*_G$.

Finally, we note that the strong relation between the complexity of GRAPH ISOMORPHISM and ISO-EQN$^*_G$ for Abelian groups that we give, presents a new approach for settling one of the longest standing open problems in complexity theory.

## References


1. M. Agrawal and T. Thierauf. The formula isomorphism problem. *SIAM Journal on Computing*, 30(3):990–1009, 2000.
2. E. Böhler, E. Hemaspaandra, S. Reith, and H. Vollmer. Equivalence and isomorphism for boolean constraint satisfaction. In *Conference for Computer Science Logic*, pages 412–426, 2002.
3. E. Böhler, E. Hemaspaandra, S. Reith, and H. Vollmer. The complexity of boolean constraint isomorphism. In *Proceedings of the 21st Symposium on Theoretical Aspects of Computer Science*, pages 164–175, 2004.
4. K.S. Booth and C.J. Colbourn. Problems polynomially equivalent to graph isomorphism. Technical report, CS-77-04, Computer Science Dept., University of Waterloo, 1979.
5. M. Goldmann and A. Russel. The complexity of solving equations over finite groups. *Information and Computation*, 178(1):253–262, 2002.
6. H. Köbler, U. Schöning, and J. Torán. *The Graph Isomorphism Problem: Its Structural Complexity*. Birkhäuser Boston, 1993.
7. R. Mathon. A note on the graph isomorphism counting problem. *Information Processing Letters*, 8(3):131–132, 1979.
8. C. Moore, P. Tesson, and D. Thérien. Satisfiability of systems of equations over finite monoids. In *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science*, pages 537–547, 2001.
9. G. Nordh and P. Jonsson. The complexity of counting solutions to systems of equations over finite semigroups. In *Proceedings of the 10th International Computing and Combinatorics Conference*, 2004.
10. P. Tesson. *Computational Complexity Questions Related to Finite Monoids and Semigroups*. PhD thesis, School of Computer Science, McGill University, Montreal, 2003.