

# An algorithm for constructing all supercharacter theories of a finite group\*

Ali Reza Ashrafi 


*Department of Pure Mathematics, Faculty of Mathematical Sciences,  
University of Kashan, Kashan 87317–53153, I. R. Iran*

Leila Ghanbari-Maman

*Department of Computer Science, Faculty of Mathematical Sciences,  
University of Kashan, Kashan 87317–53153, I. R. Iran, and  
Department of Bioinformatics, Institute of Biochemistry and Biophysics,  
University of Tehran, Tehran, Iran*

Kaveh Kavousi 

*Department of Bioinformatics, Institute of Biochemistry and Biophysics,  
University of Tehran, Tehran, Iran*

Fatemeh Koorepazan-Moftakhar 

*Department of Pure Mathematics, Faculty of Mathematical Sciences,  
University of Kashan, Kashan 87317–53153, I. R. Iran, and  
Department of Mathematical Sciences, Sharif University of Technology,  
Azadi Street, P.O. Box 11155-9415, Tehran, Iran*

Received 14 February 2019, accepted 22 December 2019, published online 15 October 2020

---

## Abstract

In 2008, Diaconis and Isaacs introduced the notion of a supercharacter theory of a finite group in which supercharacters replace with irreducible characters and superclasses by conjugacy classes. In this paper, we introduce an algorithm for constructing supercharacter theories of a finite group by which all supercharacter theories of groups containing up to 14 conjugacy classes are calculated.

*Keywords:* Supercharacter theory, superclass, conjugacy class, irreducible character.

*Math. Subj. Class. (2020):* 20C15, 20D15

---

\*The authors are indebted to an anonymous referee for his/her useful comments and suggestions that led us to rearrange this paper. We are also indebted to professor Thomas Breuer for some critical discussion on primitive groups and the method he has suggested for introducing them in GAP.

## 1 Introduction

Suppose  $UT_n(q)$  denotes the set of all  $n \times n$  unipotent upper-triangular matrices over the finite field  $GF(q)$ . While working on the complex characters of this group, André constructed something nowadays called a supercharacter theory [1, 2, 3]. Diaconis and Isaacs in their seminal paper [9], axiomatized the notion of supercharacter theories of finite groups. To define, we assume that  $G$  is a finite group,  $\text{Irr}(G)$  denotes the set of all ordinary irreducible characters of  $G$  and  $\text{Con}(G)$  is the set of all conjugacy classes of  $G$ . A pair  $(\mathcal{X}, \mathcal{K})$  is a supercharacter theory of  $G$  if the following conditions hold:

1.  $\mathcal{X}$  and  $\mathcal{K}$  are set partitions of  $\text{Irr}(G)$  and  $\text{Con}(G)$ , respectively;
2.  $\mathcal{K}$  contains  $\{e\}$ , where  $e$  denotes the identity element of  $G$ ;
3.  $|\mathcal{X}| = |\mathcal{K}|$ ;
4. For every  $X \in \mathcal{X}$ , characters  $\sigma_X = \sum_{\chi \in X} \chi(e)\chi$  are constant on each  $K \in \mathcal{K}$ .

Characters  $\sigma_X$  are called *supercharacters*, and the members of  $\mathcal{K}$  are *superclasses* of  $G$  [9]. Throughout this paper,  $\text{Sup}(G)$  denotes the set of all supercharacter theories of  $G$ . Assume  $\mathcal{X} = \{\{1_G\}, \text{Irr}(G) \setminus \{1_G\}\}$  and  $\mathcal{K} = \{\{e\}, \text{Con}(G) \setminus \{e\}\}$ . Then  $m(G) = (\text{Irr}(G), \text{Con}(G))$  and  $M(G) = (\mathcal{X}, \mathcal{K})$  are the trivial supercharacter theories of  $G$ .

We now review some constructive results on supercharacter theories of finite groups. Hendrickson [13] provided several constructions which are used to classify all supercharacter theories of cyclic groups and obtained an exact formula for the number of supercharacter theories of a finite cyclic  $p$ -group. By studying partitions of the set of irreducible characters, Clifford theory and some well-known results regarding the structure of simple rational groups, Burkett et al. [6] proved that there are only three groups with exactly two supercharacter theories: the cyclic group  $Z_3$ , the symmetric group  $S_3$  which is solvable, and the non-abelian simple group  $\text{Sp}(6, 2)$ . Furthermore, Wynn [22] described all supercharacter theories of extraspecial and Frobenius groups. The number of supercharacter theories of dihedral groups of order  $2p$ ,  $p$  is a Mersenne prime, was also calculated. In particular, he proved that if  $G$  is a Frobenius group of order  $pq$ , where  $p, q$  are primes and  $p > q$ , then  $G$  has exactly  $1 + \tau\left(\frac{p-1}{q}\right)\tau(q-1)$  supercharacter theories in which  $\tau(n)$  denotes the number of positive divisors of  $n$ . In [5] the authors continued these works by providing some constructive methods in order to find new supercharacter theories. Then, they applied these methods towards a classification of finite simple groups with exactly three or four supercharacter theories.

The aim of this paper is to present an algorithm for constructing all supercharacter theories of finite groups. To explain and then evaluate our algorithm, we need some concepts in computer science. The time complexity of a program with a given input data of size  $n$  is defined as the number of elementary instructions that this program executes as a function of  $n$ . Moreover, the space complexity of a program with a given input data of size  $n$  is defined as the number of elementary objects that this program needs to store during its execution with respect to  $n$ . For two matrices  $A$  and  $B$  with the same number of rows, the augmented matrix  $C = [A|B]$  is formed by appending the columns of  $B$  to  $A$ .

Throughout this paper, our calculations are done with the aid of GAP [12]. Our group theory notations and terminologies can be found in [14, 18]. Moreover, we refer the interested readers to the book [8] for more information on algorithms.

## 2 Algorithm

Set  $[n] = \{1, 2, \dots, n\}$ ,  $G$  is a finite group,  $\text{Irr}(G) = \{\chi_1, \dots, \chi_n\}$  and  $\text{Con}(G) = \{K_1, \dots, K_n\}$ . Choose  $A \subseteq [n]$ . Define  $A_I(G) = \{\chi_i \mid i \in A\}$  and  $A_C(G) = \{K_i \mid i \in A\}$ . Then the mappings  $\xi_1: \mathcal{P}([n]) \rightarrow \mathcal{P}(\text{Irr}(G))$  and  $\xi_2: \mathcal{P}([n]) \rightarrow \mathcal{P}(\text{Con}(G))$  are given by  $\xi_1(A) = A_I(G)$  and  $\xi_2(A) = A_C(G)$ , where  $\mathcal{P}(Y)$  denotes the power set of a given set  $Y$ . Conversely, we assume that  $B \subseteq \text{Irr}(G)$ ,  $C \subseteq \text{Con}(G)$  and define  $[n]_B = \{i \mid \chi_i \in B\}$  and  $[n]^C = \{j \mid K_j \in C\}$ . We now define  $\gamma_1: \mathcal{P}(\text{Irr}(G)) \rightarrow \mathcal{P}([n])$  and  $\gamma_2: \mathcal{P}(\text{Con}(G)) \rightarrow \mathcal{P}([n])$  by  $\gamma_1(B) = [n]_B$  and  $\gamma_2(C) = [n]^C$ . Then the mapping  $\xi_t$  and  $\gamma_t$ ,  $t = 1, 2$ , are mutually inverse and so we can use  $\mathcal{P}([n])$  instead of both  $\mathcal{P}(\text{Con}(G))$  and  $\mathcal{P}(\text{Irr}(G))$  in our algorithms.

Let  $X = \{\chi_1, \dots, \chi_u\}$  and  $K = \{K_1, \dots, K_s\}$  be parts of set partitions  $\mathcal{X}$  of  $\text{Irr}(G)$  and  $\mathcal{K}$  of  $\text{Con}(G)$ , respectively. If  $\sigma_X(K_1) = \dots = \sigma_X(K_s)$ , then we say that  $X$  and  $K$  are *consistent*. If all parts of  $\mathcal{X}$  are mutually consistent with all parts of  $\mathcal{K}$ , then the set partitions  $\mathcal{X}$  and  $\mathcal{K}$  are said to be consistent. In a part of the proof of [9, Theorem 2.2(c)], the following equivalence relation on  $G$  is given.

$$u \sim v \iff \forall X \in \mathcal{X}, \sigma_X(u) = \sigma_X(v).$$

Note that if  $u$  and  $v$  are conjugate in  $G$ , then  $u \sim v$ . As a result, it is enough to compute  $\sigma_X$  on all conjugacy classes of  $G$ . Suppose  $I = \{X_1, \dots, X_r\}$  is a set partition of  $\text{Irr}(G)$  and define  $\sigma_i = \sigma_{X_i}$ ,  $1 \leq i \leq r$ . Set  $K_x = \{y \mid \forall i, 1 \leq i \leq r; \sigma_i(x) = \sigma_i(y)\}$ ,  $x \in G$ , and let  $J$  be the set of all such subsets. If all members of  $J$  are non-empty, then  $J$  is a set partition of  $\text{Con}(G)$  consistent with  $I$  and so  $(I, J) \in \text{Sup}(G)$ . It is far from true that for each set partition  $\mathcal{X}$  of the irreducible characters of  $G$  there exists a set partition  $\mathcal{K}$  of  $G$ -conjugacy classes such that  $\mathcal{X}$  and  $\mathcal{K}$  are consistent. Hence the problem of computing supercharacter theories of  $G$  is reduced to the problem of computing all consistent pairs for  $G$ .

For the sake of completeness, we introduce here some notations in order to work with supercharacter theories of a group  $G$  in GAP. The notation  $\sigma_X(i)$  denotes the image of  $\sigma_X$  in the  $i$ -th conjugacy class of  $G$ . We also use the notation  $[n, i] = \text{SmallGroup}(n, i)$  to denote the  $i$ -th group of order  $n$  in the small group library of GAP.

The aim of this section is to present an algorithm for constructing all supercharacter theories of a finite group  $G$ . We partition this algorithm into three sub-algorithms. These sub-algorithms are presented in three sub-sections. In the first sub-section, a sub-algorithm for finding the set of all bad parts is provided. The second sub-section devotes to calculating all set partitions of an  $n$ -element set such that these set partitions do not have any bad part. In the last sub-section, a sub-algorithm for computing a consistent set partition of the conjugacy classes of a group  $G$  with respect to a set partition of  $\text{Irr}(G)$  is given.

### 2.1 Bad parts and bad set partitions

A part  $X$  of a set partition  $\mathcal{X}$  of  $\text{Irr}(G)$  is said to be *bad* if  $X$  is consistent with only singleton subsets of  $\text{Con}(G)$ . A set partition containing a bad part is called a *bad set*

partition. It is easy to see that a bad set partition  $\mathcal{X}$  of  $\text{Irr}(G)$  does not have a mate  $\mathcal{K}$  such that  $(\mathcal{X}, \mathcal{K}) \in \text{Sup}(G)$ .

We recall that in computing supercharacter theories of a finite group  $G$ ,  $\{e\}$  and  $\{1_G\}$  are always parts of  $\mathcal{K}$  and  $\mathcal{X}$ , respectively. As a result, it is enough to work with the set partitions of  $[n]^* = \{2, \dots, n\}$ .

**Lemma 2.1.** *Suppose  $X \in \mathcal{P}([n]^*)$  is a bad part. Then all values of  $\sigma_X(i)$ ,  $2 \leq i \leq n$ , are distinct.*

*Proof.* Choose  $2 \leq j \neq k \leq n$  such that  $\sigma_X(j) = \sigma_X(k)$ . Then the part  $X$  is consistent with  $\{j\}$ ,  $\{k\}$  and  $\{j, k\}$ . This is a contradiction to the definition of a bad part.  $\square$

By Lemma 2.1, to check whether a part  $X \in \mathcal{P}([n]^*)$  is bad or not, it is enough to compute  $\sigma_X(i)$  for  $i \in [n]^*$ . If all values are different, then  $X$  is a bad part.

### 2.2 Create set partitions

The computer algebra system GAP does not have the potential to construct set partitions for large enough  $[n]$ . In literature, there are two algorithms by Semba [17] and Er [11] for computing set partitions of  $[n]$ . The Semba’s algorithm is based on the backtrack technique [17, Theorem 1] and has the time complexity  $\Theta(4B(n))$ , where the Bell number  $B(n)$  is defined as the number of set partitions of an  $n$ -element set. The Er’s algorithm is recursive. He claimed (without proof) that  $\sum_{i=1}^n B(i) < 1.6B(n)$ . This is while Nayak and Stojmenović [16, p. 12] proved that  $\sum_{i=2}^n B(i) < 2B(n)$ . In an exact phrase, Er claimed that the time complexity of his algorithm for generating all set partitions of  $[n]$  is  $\Theta(1.6B(n))$ .

There exists a limitation in the Er’s algorithm: Each set partition is determined at the end step and so we cannot identify whether a given set partition is bad or not, unless it is done completely. As a consequence, we design an algorithm which generates set partitions part by part. When a bad part occurs, calculations of all set partitions containing that part are pruned.

To explain our algorithm, we set  $S = \{s_1, s_2, \dots, s_n\}$ . The  $2^n - 1$  non-empty subsets of  $S$  are used as the parts of the set partitions of  $S$ . Note that when  $n$  is enough large, it is not possible to save all set partitions on the memory. In order to reduce memory usage, we use the integers of the closed interval  $I = I(S) = [1, 2^{|S|} - 1]$ . In fact, we define a one-to-one correspondence  $\alpha_S: I \rightarrow \mathcal{P}(S) \setminus \{\emptyset\}$  by  $\alpha_S(k) = \{s_i \mid a_{n+1-i} = 1\}$ , where  $a_1 a_2 \dots a_n$  is the  $n$ -bit binary form of  $k$ . For example, if  $S = \{s_1, s_2, s_3, s_4, s_5\}$  and  $k = 13$  then  $(13)_2 = 1101$  and since  $|S| = 5$ , the 5-bit binary form of 13 is 01101. Thus,  $\alpha_S(13) = \{s_1, s_3, s_4\}$ . Let  $I_O = I_O(S)$  be the set of all odd integers in the closed interval  $I$ . Then  $\alpha_S(I_O)$  is the set of all subsets of  $S$  containing  $s_1$ . On the other hand, if  $F$  is a non-empty subset of  $S$ , then we conclude that  $\alpha_S^{-1}(F) = \sum_{i \in F} 2^{\text{Position}(S, i) - 1}$ , where  $\text{Position}(S, i)$ ,  $i \in F$ , is the position of  $i$  in  $S$ . In this example, if  $F = \{s_1, s_3, s_4\}$  then  $\alpha_S^{-1}(F) = 2^{1-1} + 2^{3-1} + 2^{4-1} = 13$ .

Now, we are ready to present our algorithm for constructing all set partitions with no bad part. We use two lists  $SPs$  and  $RE$  in order to keep the set partitions and remaining elements, respectively. At the first step, we have  $SPs = \emptyset$  and  $RE = [n]^*$ . We fill  $SPs$  by parts constructed from the elements of  $RE$  which are not bad. Thus,  $RE := RE \setminus \alpha_{RE}(k)$  and  $SPs := SPs \cup \{\alpha_{RE}(k)\}$ , where  $k \in I_O$  and  $\sigma_{RE}(k)$  is not a bad part. This algorithm will be returned to the previous step, when  $RE = \emptyset$ . Since our algorithm is recursive, its generating tree is constructed by DFS strategy. In Sub-algorithm 2, all set partitions of  $[n]^*$

that do not contain any bad part are generated. In Figure 1, an example of a generating tree for set partitions of  $[4]^*$  is presented.

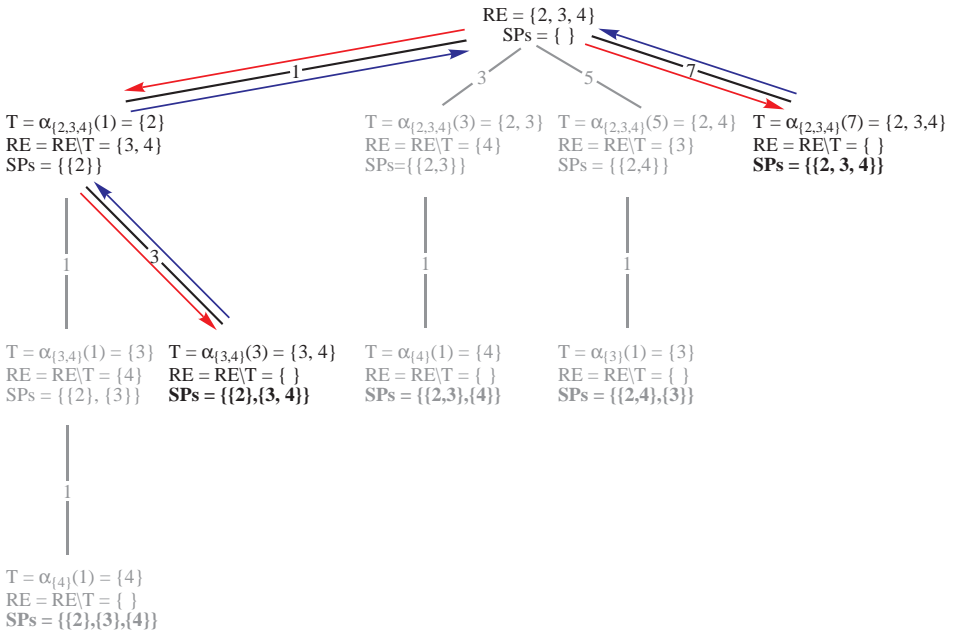


Figure 1: A schematic diagram of Sub-algorithm 2 for the case that  $RE = [4]^*$  and badparts =  $\{\{2, 3\}, \{2, 4\}, \{3\}, \{4\}\}$ . The red and blue arrows represent forward and backward directions in the generating tree traversal, respectively. The number on each edge is an odd integer in  $I_O(RE)$  of the parent node of the edge. The gray part of the tree is pruned due to the occurrence of a bad part in the process of creating corresponding set partition.

Note that after creating a set partition for  $\text{Irr}(G)$ , we invoke the Sub-algorithm 3 for it, to check whether there exists a consistent set partition of  $\text{Con}(G)$  or not. This function is explained in details in the next subsection.

### 2.3 Create the consistent set partition $\mathcal{K}$ with respect to a given set partition of $\text{Irr}(G)$

We recall that finding all supercharacter theories of a group  $G$  with  $n$  conjugacy classes is equivalent to constructing all consistent pairs of set partitions. Suppose  $\text{Irr}p$  is a given set partition of the irreducible characters of  $G$ . To find a consistent set partition of  $\text{Irr}p$ , we first define the matrix  $A$  as follows, see Table 1.

- The rows of  $A$  are the parts of  $\text{Irr}p$  and so  $A$  has exactly  $|\text{Irr}p|$  rows.
- The columns of  $A$  are the conjugacy classes of  $G$ .
- If  $A = (a_{ij})$ , then  $a_{ij} = \sigma_{X_i}(K_j)$  where  $1 \leq i \leq |\text{Irr}p|$ ,  $1 \leq j \leq n$  and  $K_j$  are the conjugacy classes of  $G$ .

Suppose  $C_1, C_2, \dots, C_n$  are all columns of  $A$ . We construct a matrix  $ST$  and a list  $Kappa$  as follows. Since  $\{e\}$  is a part of each consistent set partition with  $Irrp$ , we conclude that  $\{1\} \in Kappa$ . We start our algorithm by defining  $Kappa = \{\{1\}, \{2\}\}$  and the submatrix  $ST = [C_1, C_2]$ . For each  $j, 3 \leq j \leq n$ , we compare  $C_j$  with all constructed columns of  $ST$  other than its first column. If  $C_j$  is different from such columns of  $ST$ , then we add  $C_j$  to  $ST$  as a new column and add  $j$  to  $Kappa$  as a singleton part. Hence  $Kappa := Kappa \cup \{\{j\}\}$  and  $ST := [ST|C_j]$ . If  $C_j$  is equal to the  $r$ -th column of  $ST$ , then we add  $j$  to the  $r$ -th part of  $Kappa$ , i.e.

$$Kappa = \{\{1\}, \dots, \underbrace{\{\dots, j\}}_{\text{part } r}, \dots, \{\dots\}\}.$$

Table 1: Matrix  $A$ .

	$K_1$	$K_2$	$K_3$	$\dots$	$K_n$
$\chi_1$	1	1	1	$\dots$	1
$\vdots$	*	*	*	$\dots$	*
$X_i \begin{cases} \chi_{i_1} \\ \vdots \\ \chi_{i_t} \end{cases}$	$\sigma_{X_i}(K_1)$	$\sigma_{X_i}(K_2)$	$\sigma_{X_i}(K_3)$	$\dots$	$\sigma_{X_i}(K_n)$
$\vdots$	*	*	*	$\dots$	*

If in the process of constructing  $Kappa$  and  $ST$  the inequality  $|Kappa| > |Irrp|$  occurs, then we stop calculations without any result. It is because there is no consistent set partition with the same size as  $Irrp$ . If at the end of our calculations,  $|Kappa| = |Irrp|$ , then we conclude that  $(Irrp, Kappa)$  is a supercharacter theory and  $ST$  is the supercharacter table of  $G$ .

To construct all supercharacter theories of a group  $G$ , it is possible to combine the Er’s algorithm which creates all set partitions of  $Irr(G)$  with the algorithm based on the proof of Theorem 2.2(c) in [9]. This is our *first algorithm*. Our main algorithm is a combination of the Sub-algorithms 1, 2 and 3. In [4], Sub-algorithm 3 for computing bad parts of the cyclic group  $Z_{13}$  is analyzed.

We end this section by noticing that:

1. The result of our main algorithm is supercharacter theories of a given group  $G$ . In fact, conditions of being a supercharacter theory are checked by the main algorithm in each case.
2. All supercharacter theories are generated by our algorithm. This is guaranteed by the proof of [9, Theorem 2.2(c)].

### 3 Analysis of algorithms

In Section 2, three sub-algorithms for computing bad parts, set partitions and supercharacter theories were presented. The aim of this section is to calculate the running time and the space complexity of these sub-algorithms and also our main algorithm.

**Theorem 3.1.** Let  $T_1(n)$ ,  $S_1(n)$  and  $BP$  be the time complexity function, the space complexity function and the list of bad parts for a given group, respectively. Then,  $T_1(n) \in O((n^2 - n) \cdot 2^{n-1})$  and  $S_1(n) \in O(n) + O(|BP|)$ , where  $n = |\text{Con}(G)|$ .

*Proof.* To compute the running time of the Sub-algorithm 2, we should know the values of  $\sigma_X(j)$ ,  $2 \leq j \leq n$ , where  $X = \{x_1, \dots, x_i\} \in \mathcal{P}([n]^*)$ . For this purpose,  $i(n-1)$  multiplications and  $(i-1)(n-1)$  additions are needed. Therefore, we have  $\frac{(n-1)(n-2)}{2}$  comparisons for investigating the property that  $\sigma_X$ 's are distinct. Since there are  $\binom{n-1}{i}$   $i$ -subsets, the complexity of this sub-algorithm can be computed by the following formula:

$$T_1(n) = \sum_{i=1}^{n-1} \left[ (n-1)(2i-1) \binom{n-1}{i} + \frac{(n-1)(n-2)}{2} \right].$$

Therefore,

$$\begin{aligned} T_1(n) &= \sum_{i=1}^{n-1} \left[ (n-1)(2i-1) \binom{n-1}{i} + \frac{(n-1)(n-2)}{2} \right] \\ &= (n-1) \sum_{i=1}^{n-1} (2i-1) \binom{n-1}{i} + \frac{(n-1)^2(n-2)}{2} \\ &< n \cdot \sum_{i=1}^{n-1} 2i \cdot \binom{n-1}{i} + n^3 = 2n \cdot \sum_{i=1}^{n-1} i \cdot \binom{n-1}{i} + n^3 \\ &= 2n \cdot (n-1) \cdot 2^{n-2} + n^3 = (n^2 - n) \cdot 2^{n-1} + n^3. \end{aligned}$$

Hence  $T_1(n) \in O((n^2 - n) \cdot 2^{n-1})$ .

To compute the space complexity of this sub-algorithm, we note that all parts are generated one by one. If a generated part is bad, then we add it to  $BP$ . To keep each part, our calculations need an array of size  $n-1$ . Moreover, an  $(n-1)$ -length array is needed in order to save the values  $\sigma_X(i)$ . Therefore, this sub-algorithm needs a memory of size  $O(n)$  to keep each part. For saving all bad parts, we need another array such that its size depends only on the number of bad parts of irreducible characters of a given group. Consequently,  $S_1(n) \in O(n) + O(|BP|)$ .  $\square$

**Theorem 3.2.** Suppose  $T_2(n)$  and  $S_2(n)$  are the time and space complexity functions of the Sub-algorithm 2, respectively. Then,

- (1)  $T_2(n) \in O(2B(n))$ ;
- (2) The space complexity  $S_2(n)$  belongs to  $\Theta(n)$ .

*Proof.* To prove (1), let  $T_2(n)$  denote the number of calculations needed to obtain all set partitions which do not contain bad parts and are from the  $(n-1)$ -element set  $RE$ . For computing the time complexity in the worst case  $|BP| = 0$ , we have to count the number of edges in the generating tree of this sub-algorithm in general, see Figure 1. Then,

$$T_2(n) = \sum_{i=0}^{n-1} \binom{n-1}{i} (T_2(i) + 1).$$

In OEIS [10], the sequence  $\{a(n)\}_{n \geq 0}$  with code A060719 exists which is defined as follows.

$$a(n + 1) = a(n) + \sum_{i=0}^n \binom{n}{i} (a(i) + 1); \quad a(0) = 1$$

By [15],  $a(n) = 2B(n + 1) - 1$  and since  $T_2(n) = a(n - 1)$ , we conclude that  $T_2(n) = 2B(n) - 1$ . Hence, the time complexity of this algorithm is  $O(2B(n))$ .

The space complexity depends on the sizes of *SPs* and *RE*. Since the union of *RE* with the members of *SPs* is  $[n]$ ,  $S_2(n) \in \Theta(n)$  which proves (2). □

In the next theorem, we calculate the complexity of our Sub-algorithm 3 which computes the supercharacter theories of  $G$ .

**Theorem 3.3.** *Suppose  $G$  has exactly  $n$  conjugacy classes. For a given set partition  $Irrp$ , the time and space complexities of the Sub-algorithm 3 are  $O(n^3)$  and  $O(n^2)$ , respectively.*

*Proof.* Suppose  $Irrp$  is a set partition for the set of all irreducible characters of a group  $G$  and  $|Irrp| = k$ . To calculate the matrix  $A$ , we first obtain all values  $\chi_i(1)\chi_i$ ,  $1 \leq i \leq n$ . Since  $\chi_i$  has  $n$  values, there are  $n^2$  different products  $\chi_i(1)\chi_i$ . To compute  $\sigma_X$  and in the worst case  $X = \{\chi_1, \dots, \chi_n\}$ , we need  $n(n - 2)$  products. As a result, the calculations for obtaining the matrix  $A$  is of the time complexity  $O(n^2)$ .

Now, we count all the operations that we need to construct  $ST$  and the list  $Kappa$ . The first and second columns of  $ST$  are the same as the first and second columns of  $A$ , respectively. Therefore, we have nothing to count for these columns. For the third column of  $ST$ , we have to compare the third column of  $A$  with the second column of  $ST$  and so there are  $k$  comparisons. For the fourth column of  $ST$ , the fourth column of  $A$  should be compared with the second and third columns of  $ST$ , and so there are at most  $2k$  comparisons, and so on. Suppose that from the column  $C_j$  to the next,  $|Kappa| = |Irrp|$ . In this case, the remaining conjugacy classes of the group should be distributed among the other parts of  $Kappa$  and hence we do not have a new part in  $Kappa$ . Thus from  $C_j$  to  $C_n$ , the number of comparisons is equal to  $k(k - 1)$ . Therefore, the total number of comparisons for constructing  $ST$  and  $Kappa$  is:

$$T_3(n) := k(1) + k(2) + \dots + \underbrace{k(k - 1) + k(k - 1) + \dots + k(k - 1)}_{n-j+1}.$$

Since  $j \geq 3$ ,  $n - j + 1 \leq n - 2$ . Thus  $T_3(n) \leq k(\frac{n^2-5n+6}{2}) \leq n(\frac{n^2-5n+6}{2})$ , and so,  $T_3(n) \in O(n^3)$ .

Suppose  $S_3(n)$  is the space complexity of the Sub-algorithm 3. We have two matrices  $A$  and  $ST$  of sizes  $k \times n$  and  $k \times k$ , respectively. Since in the worst case  $k = n$ , we conclude that  $S_3(n) \in O(n^2)$ . □

We are now ready to compute the time and space complexity of the first and main algorithms. In the first algorithm, the generated set partitions are used as an input to compute all supercharacter theories of a finite group. In what follows, we assume that our group has exactly  $n$  conjugacy classes. The running time of our first algorithm is  $T_{Er}(n) = O(n^3) \cdot \Theta(2B(n))$  and so  $T_{Er}(n) \in O(n^3B(n))$ .

In the main algorithm, we do not need to create consistent set partition  $Kappa$  for bad set partitions. As a result, the Sub-algorithm 3 should be called  $B(n) - |BP_s|$  times, where



$|BP_s|$  is the number of bad set partitions. Therefore, the time complexity of the main algorithm is  $O((n^2 - n)2^{n-1} + (B(n) - |BP_s|)n^3) = O(n^3(B(n) - |BP_s|))$  and so the following result holds.

**Theorem 3.4.** *The time complexity of our first and main algorithms are  $O(n^3B(n))$  and  $O(n^3(B(n) - |BP_s|))$ , respectively.*

## 4 Performance evaluation

To evaluate the performance of the main algorithm and then compare it with the first one, both algorithms have been implemented in the computer algebra system GAP under Windows 10 Home Single Language. The average running times for both algorithms after three runs on a computer with processor Intel(R) Core(TM) m7-6Y75 CPU @ 1.20 GHz 1.51 GHz, installed memory (RAM) 8.00 GB (7.90 GB usable), system type 64-bit operating system and x64-based processor are summarized in Table 2. In this table, we have chosen groups which have the maximum or the minimum number of supercharacter theories with different number of conjugacy classes. Let  $BP(G)$  be the set of all bad parts in a group  $G$  and  $\alpha(G) = \frac{|BP(G)|}{2^{\kappa(G)-1}-1}$  in which  $\kappa(G)$  denotes the number of distinct conjugacy classes of  $G$ . We have the following two cases in general.

1. **There is not any bad part in  $\mathcal{P}(\text{Irr}(G))$ .** In this case, the algorithm for computing supercharacter theories based on the Er's algorithm has a faster running time. Note that we have a pre-process for finding bad parts but such an overhead is very small with respect to the total running time.
2. **There are some bad parts in  $\mathcal{P}(\text{Irr}(G))$ .** In this case, by removing these parts from our calculations, the main algorithm will have a faster running time. For example, in the cyclic group  $Z_{13}$  in which %98.17 of all parts are bad, our main algorithm takes less than one second to run while the other algorithm takes more than 548 seconds. In rare cases such as the Mathieu group  $M_{22}$  in which a few percentage of existence of parts are bad, the running time of the first algorithm is a bit faster.

To compare the first algorithm with the main one, we use Figure 2 in which the running time of the groups with exactly 13 conjugacy classes with respect to these algorithms are shown. As it can be seen, groups numbered 29–53 have faster running times with the main algorithm. The result shows that the main algorithm is better for some classes of groups.

## 5 Concluding remarks

In this paper, two algorithms for computing all supercharacter theories of a finite group  $G$  have been presented. The first algorithm is based on the Er's algorithm. In the main algorithm, we have introduced the new feature “bad part” for the parts of  $\text{Irr}(G)$ . Since none of the supercharacter theories contains these bad parts, by filtering and detecting the set partitions of  $\text{Irr}(G)$  which have at least one bad part, the running time of this algorithm decreases significantly.

Suppose  $BP(G)$  denotes the set of all bad parts in a group  $G$ . Let  $\alpha(G) = \frac{|BP(G)|}{2^{\kappa(G)-1}-1}$  and assume  $|\text{Sup}(G)|$  is the number of all supercharacter theories of  $G$ . In Table 4, the percentage of existence of bad parts for some cyclic and dihedral groups are given.

Table 2: Comparing the running times for some groups.

$\kappa(G)$	$G$	$ \text{Sup}(G) $	$ \text{BP}(G) $	$\alpha(G)$	Main algorithm (MA)(second)	First algorithm (FA)(second)	FA/MA
10	[100, 11]	623	0	0	1.4	1.1	0.8
11	[32, 43]	376	0	0	7.6	6.7	0.9
11	[32, 44]	376	0	0	8.1	7.5	0.9
12	[1296, 3523]	1058	0	0	70.1	62	0.9
13	[64, 32]	325	0	0	464.6	429.8	0.9
12	$D_{36}$	51	168	8.2	65.4	68.2	1.04
12	$M_{22}$	5	288	14.1	65.8	61.8	0.9
10	$M_{11}$	5	112	21.9	0.8	1.1	1.4
10	$D_{28}$	23	144	28.9	0.8	1.7	2.1
10	[120, 35]	10	152	29.7	0.6	1.1	1.8
13	[93, 1]	9	1980	48.4	169.7	662.3	3.9
13	[253, 1]	9	1980	48.4	127.8	521.3	4.08
10	$Z_{10}$	10	376	73.6	0.06	1.2	20
10	$D_{34}$	5	480	93.9	0.04	1.3	32.5
11	$Z_{11}$	4	990	96.8	0.1	7.9	79
13	$Z_{13}$	6	4020	98.1	1.2	548.2	456.8
11	$D_{38}$	4	1008	98.5	0.1	8.5	85
13	$D_{46}$	3	4092	99.9	1.2	610.6	508.8

Table 3: The GAP id of all groups with exactly 13 conjugacy classes.

1	[162, 21]	12	[162, 20]	23	[960, 11359]	34	[100, 10]	45	[328, 12]
2	[96, 191]	13	[1944, 2290]	24	[64, 33]	35	[162, 15]	46	[148, 3]
3	[400, 206]	14	[64, 37]	25	[1000, 86]	36	[40, 6]	47	[333, 3]
4	[162, 22]	15	[64, 32]	26	[720, 409]	37	[1053, 51]	48	[156, 7]
5	[192, 1494]	16	[96, 190]	27	[576, 8652]	38	[120, 38]	49	[301, 1]
6	[96, 193]	17	[192, 1491]	28	[40, 4]	39	[600, 148]	50	[205, 1]
7	[1944, 2289]	18	[64, 36]	29	<b>[162, 11]</b>	40	[216, 86]	51	[150, 5]
8	[192, 1493]	19	[192, 1492]	30	[160, 199]	41	[258, 1]	52	[13, 1]
9	[1440, 5841]	20	[64, 35]	31	[324, 160]	42	[310, 1]	53	[46, 1]
10	[40, 8]	21	[162, 19]	32	[162, 13]	43	[253, 1]		
11	[64, 34]	22	[216, 87]	33	[1320, 133]	44	[93, 1]		

Our calculations given in Table 4 suggests the following conjecture.

**Conjecture 5.1.** *If  $\beta_n = \alpha(Z_{p_n})$  and  $\gamma_n = \alpha(D_{2p_n})$ , then*

$$\lim_{n \rightarrow \infty} \beta_n = \lim_{n \rightarrow \infty} \gamma_n = 1,$$

where  $p_n$  is the  $n$ -th prime number.

In Table 5, the percentage of existence of bad parts in some groups of order  $3p, 3 \mid p - 1$  is presented. The calculations given in this table show that the Conjecture 5.1 is not valid for groups of order  $3p$ . Suppose  $p$  and  $q$  are primes such that  $q < p$  and  $q \mid p - 1$ . Let

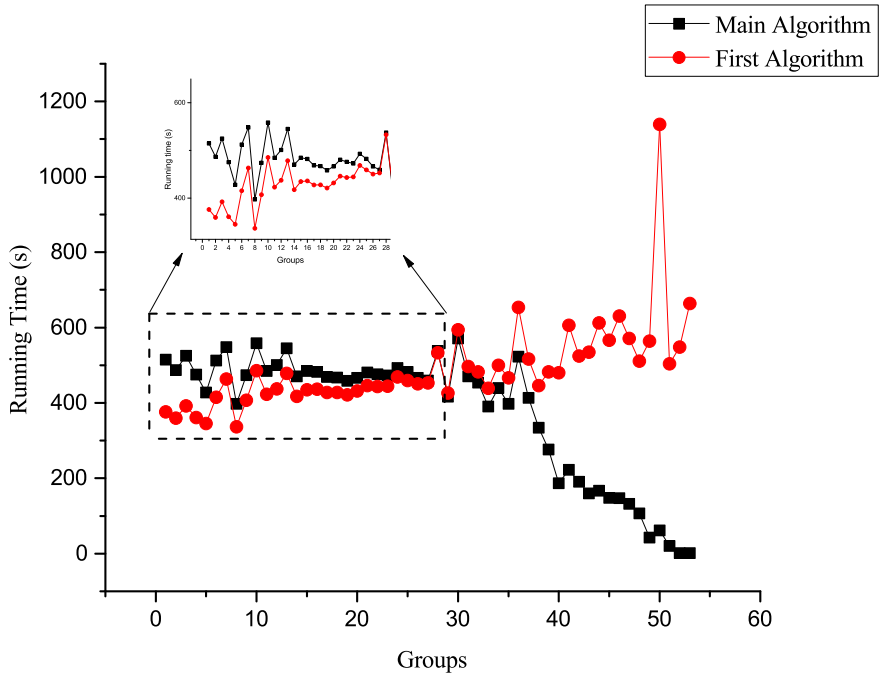


Figure 2: A diagram for the running time of all 53 groups which are listed in Table 3.

Table 4: Percentage of existence of bad parts for some cyclic and dihedral groups.

$\kappa(G)$	Group	$\alpha(G)$	$ \text{Sup}(G) $	$\kappa(G)$	Group	$\alpha(G)$	$ \text{Sup}(G) $
4	$D_4$	% 0	5	2	$Z_2$	% 100	1
3	$D_6$	% 66.67	2	3	$Z_3$	% 66.67	2
4	$D_{10}$	% 57.14	3	5	$Z_5$	% 80	3
5	$D_{14}$	% 80	3	7	$Z_7$	% 85.7	4
7	$D_{22}$	% 95	3	11	$Z_{11}$	% 96.77	4
8	$D_{26}$	% 84.3	5	13	$Z_{13}$	% 98.16	6
10	$D_{34}$	% 93.75	5	17	$Z_{17}$	% 99.6	5
11	$D_{38}$	% 98.53	4	19	$Z_{19}$	% 99.78	6
13	$D_{46}$	% 99.92	3				
16	$D_{58}$	% 99.2	5				
17	$D_{62}$	% 99.88	5				

$T_{p,q}$  denote the non-abelian group of order  $pq$  and  $q_n$  denote the  $n$ -th prime number with the property  $3 \mid q_n - 1$ .

Table 5: Percentage of existence of bad parts in  $T_{p,3}$ .

$p$	$\alpha(G)$	$p$	$\alpha(G)$	$p$	$\alpha(G)$
7	% 26	13	% 38	19	% 42
31	% 48	37	% 49	43	% 49.6

By the results of Table 5, we offer the following conjecture:

**Conjecture 5.2.** *If  $\delta_n = \alpha(T_{q_n,3})$ , then  $\lim_{n \rightarrow \infty} \delta_n = 0.5$ .*

Table 6: Supercharacter theory form of groups with  $\kappa \leq 14$  conjugacy classes.

$\kappa$	Supercharacter Theory Form
3	$2^2$
4	$3^3 5^1$
5	$3^3 5^3 9^2$
6	$4^1 5^1 7^1 8^1 9^1 (15)^1 (18)^1 (20)^1$
7	$3^1 4^1 5^1 7^3 8^2 (11)^1 (20)^3$
8	$5^2 7^1 (10)^1 (11)^1 (12)^3 (13)^1 (14)^1 (15)^1 (16)^1 (18)^1 (19)^1 (22)^1 (23)^1 (25)^1 (28)^1 (54)^1 (100)^1 (110)^1$
9	$3^1 7^3 9^1 (10)^1 (12)^1 (13)^1 (15)^1 (18)^1 (19)^1 (21)^1 (22)^2 (32)^2 (36)^1 (43)^1 (40)^1 (45)^3 (49)^1 (65)^1 (128)^2$
10	$5^2 (10)^2 (11)^1 (13)^1 (14)^1 (15)^1 (16)^1 (23)^1 (25)^2 (23)^1 (24)^1 (28)^1 (32)^2 (34)^1 (35)^2 (44)^1 (51)^1 (52)^1 (57)^2 (58)^3 (64)^2 (80)^1 (83)^2 (165)^1 (215)^2 (623)^1$
11	$4^2 8^2 (11)^4 (13)^4 (15)^1 (17)^1 (18)^2 (25)^1 (26)^1 (31)^1 (47)^3 (53)^2 (55)^1 (81)^1 (89)^2 (124)^1 (144)^3 (232)^1 (376)^2$
12	$5^1 7^1 (13)^2 (16)^1 (18)^2 (19)^2 (22)^3 (23)^2 (32)^2 (34)^1 (35)^1 (36)^1 (46)^1 (49)^1 (51)^1 (65)^1 (68)^1 (69)^1 (76)^3 (81)^2 (88)^2 (94)^1 (99)^2 (100)^1 (105)^1 (133)^4 (144)^1 (152)^1 (197)^1 (205)^1 (212)^1 (233)^1 (255)^1 (360)^1 (484)^1 (1058)^1$
13	$3^1 6^1 9^2 (11)^1 (13)^2 (17)^1 (18)^1 (24)^2 (25)^2 (35)^1 (38)^2 (40)^2 (42)^1 (43)^1 (46)^2 (50)^1 (53)^2 (71)^3 (72)^2 (81)^2 (89)^1 (102)^1 (110)^4 (129)^4 (132)^3 (138)^1 (175)^1 (313)^3 (325)^3$
14	$5^2 9^1 (10)^1 (12)^1 (13)^1 (14)^1 (15)^2 (21)^1 (22)^2 (23)^1 (29)^1 (35)^2 (38)^1 (39)^1 (41)^1 (43)^1 (45)^3 (47)^1 (49)^1 (51)^1 (53)^1 (57)^1 (63)^2 (71)^1 (76)^1 (78)^2 (79)^1 (81)^2 (85)^1 (105)^1 (110)^2 (119)^1 (123)^1 (125)^1 (130)^1 (138)^1 (139)^1 (140)^2 (145)^1 (157)^1 (172)^2 (186)^2 (206)^1 (213)^3 (222)^1 (244)^2 (270)^1 (272)^2 (304)^2 (308)^2 (320)^1 (482)^1 (601)^2 (613)^2 (620)^3 (627)^1 (645)^3 (904)^1 (940)^3 (1048)^2 (1324)^3 (2093)^1 (29016)^1$

Suppose  $\text{Irr}(Z_7) = \{1_{Z_7}, \chi_2, \dots, \chi_7\}$  and  $\text{Con}(Z_7) = \{e, x_2^{Z_7}, \dots, x_7^{Z_7}\}$ . The cyclic group  $Z_7$  has exactly four supercharacter theories  $m(Z_7), M(Z_7), \mathcal{C}_1 = (\mathcal{X}_1, \mathcal{K}_1)$  and

$\mathcal{C}_2 = (\mathcal{X}_2, \mathcal{K}_2)$  such that

$$\begin{aligned}\mathcal{X}_1 &:= \{\{1_{Z_7}\}, \{\chi_2, \chi_3, \chi_5\}, \{\chi_4, \chi_6, \chi_7\}\}, \\ \mathcal{K}_1 &:= \{\{e\}, \{x_2^{Z_7}, x_3^{Z_7}, x_5^{Z_7}\}, \{x_4^{Z_7}, x_6^{Z_7}, x_7^{Z_7}\}\}, \\ \mathcal{X}_2 &:= \{\{1_{Z_7}\}, \{\chi_2, \chi_7\}, \{\chi_3, \chi_6\}, \{\chi_4, \chi_5\}\}, \\ \mathcal{K}_2 &:= \{\{e\}, \{x_2^{Z_7}, x_7^{Z_7}\}, \{x_3^{Z_7}, x_6^{Z_7}\}, \{x_4^{Z_7}, x_5^{Z_7}\}\}.\end{aligned}$$

This shows that each part in  $\mathcal{X}$  and  $\mathcal{K}$  in a supercharacter theory  $(\mathcal{X}, \mathcal{K})$  has size 1, 2, 3 or 6. On the other hand, if  $p$  is prime and  $d$  is the number of divisors of  $p - 1$ , then by [13, Table 1], the cyclic group  $Z_p$  has exactly  $d$  supercharacter theories. As a result, the following conjecture is suggested.

**Conjecture 5.3.** *For each divisor  $r$  of  $p - 1$ , there exists only one supercharacter theory  $(\mathcal{X}, \mathcal{K})$  of  $Z_p$  such that the sizes of all non-trivial parts of  $\mathcal{X}$  and  $\mathcal{K}$  are equal to  $r$ . Moreover, if we sort the conjugacy classes and irreducible characters of  $Z_p$  by ATLAS notations [7], then  $\gamma_1(\mathcal{X}) = \gamma_2(\mathcal{K})$ .*

It is a well-known result in group theory that for any positive integer  $k$ , there are finitely many number of non-isomorphic finite groups with exactly  $k$  conjugacy classes. This number is denoted by  $f(k)$ . The supercharacter theory form of  $f(k)$  is defined as  $n_1^{\alpha_1} n_2^{\alpha_2} \cdots n_s^{\alpha_s}$  where  $\alpha_i, 1 \leq i \leq s$ , denote the number of groups with exactly  $k$  conjugacy classes containing  $n_i$  supercharacter theories and note that  $f(k) = \sum_{i=1}^s \alpha_i$ . The supercharacter theory form of groups with at most 14 conjugacy classes are recorded in Table 6 and the following conjecture has been suggested based on this table.

**Conjecture 5.4.** *The number of supercharacter theories of all members of  $\Gamma(k)$  are distinct if and only if  $k = 6$ . In this case, all groups are  $Z_5, D_{14}, A_5, Z_5 : Z_4, Z_7 : Z_3, S_4, D_8$  and  $Q_8$ .*

Vera-López and his co-authors [19, 20, 21] classified all finite groups containing up to 14 conjugacy classes. We apply these classification theorems and our main algorithm to find all supercharacter theories of groups containing up to 14 conjugacy classes. These calculations are presented in [4].

## ORCID iDs

Ali Reza Ashrafi  <https://orcid.org/0000-0002-2858-0663>

Kaveh Kavousi  <https://orcid.org/0000-0002-1906-3912>

Fatemeh Koorepazan-Moftakhar  <https://orcid.org/0000-0003-3531-5355>

## References

- [1] C. A. M. André, Basic characters of the unitriangular group, *J. Algebra* **175** (1995), 287–319, doi:10.1006/jabr.1995.1187.
- [2] C. A. M. André, Irreducible characters of finite algebra groups, in: *Matrices and Group Representations*, Universidade de Coimbra, Coimbra, volume 19 of *Texts in Mathematics, Series B*, pp. 65–80, 1999, papers from the workshop held in honor of Graciano N. de Oliveira on the occasion of his 60th birthday at the University of Coimbra, Coimbra, May 6 – 8, 1998.
- [3] C. A. M. André, Basic characters of the unitriangular group (for arbitrary primes), *Proc. Amer. Math. Soc.* **130** (2002), 1943–1954, doi:10.1090/s0002-9939-02-06287-1.

- [4] A. R. Ashrafi, L. Ghanbari Maman, K. Kavousi and F. Koorepazan Moftakhar, An algorithm for constructing all supercharacter theories of a finite group, [arXiv:1911.12232](https://arxiv.org/abs/1911.12232) [math.GR].
- [5] A. R. Ashrafi and F. Koorepazan-Moftakhar, Towards the classification of finite simple groups with exactly three or four supercharacter theories, *Asian-Eur. J. Math.* **11** (2018), 1850096 (21 pages), doi:10.1142/s1793557118500961.
- [6] S. Burkett, J. Lamar, M. L. Lewis and C. Wynn, Groups with exactly two supercharacter theories, *Comm. Algebra* **45** (2017), 977–982, doi:10.1080/00927872.2016.1172622.
- [7] J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker and R. A. Wilson, *Atlas of Finite Groups*, Oxford University Press, Eynsham, 1985.
- [8] T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, The MIT Electrical Engineering and Computer Science Series, MIT Press, Cambridge, MA, 1990.
- [9] P. Diaconis and I. M. Isaacs, Supercharacters and superclasses for algebra groups, *Trans. Amer. Math. Soc.* **360** (2008), 2359–2392, doi:10.1090/s0002-9947-07-04365-6.
- [10] F. Ellermann, Sequence A060719 in The On-Line Encyclopedia of Integer Sequences, published electronically at <https://oeis.org>.
- [11] M. C. Er, A fast algorithm for generating set partitions, *Comput. J.* **31** (1988), 283–284, doi:10.1093/comjnl/31.3.283.
- [12] The GAP Group, *GAP – Groups, Algorithms, and Programming*, <https://www.gap-system.org>.
- [13] A. O. F. Hendrickson, *Supercharacter theories of cyclic  $p$ -groups*, Ph.D. thesis, The University of Wisconsin–Madison, 2008, <https://www.proquest.com/docview/304449301>.
- [14] I. M. Isaacs, *Character Theory of Finite Groups*, Dover Publications, New York, 1994.
- [15] G. Kilibarda and V. Jovović, Antichains of multisets, *J. Integer Seq.* **7** (2004), Article 04.1.5 (15 pages), <https://cs.uwaterloo.ca/journals/JIS/VOL7/Kilibarda/kili2.html>.
- [16] A. Nayak and I. Stojmenović (eds.), *Handbook of Applied Algorithms: Solving Scientific, Engineering and Practical Problems*, Wiley-Interscience, Hoboken, NJ, 2008, doi:10.1002/9780470175668.
- [17] I. Semba, An efficient algorithm for generating all partitions of the set  $\{1, 2, \dots, n\}$ , *J. Inform. Process.* **7** (1984), 41–42.
- [18] B. Steinberg, *Representation Theory of Finite Groups: An Introductory Approach*, Universitext, Springer, New York, 2012, doi:10.1007/978-1-4614-0776-8.
- [19] A. Vera-López and J. Sangroniz, The finite groups with thirteen and fourteen conjugacy classes, *Math. Nachr.* **280** (2007), 676–694, doi:10.1002/mana.200410508.
- [20] A. Vera López and J. Vera López, Classification of finite groups according to the number of conjugacy classes, *Israel J. Math.* **51** (1985), 305–338, doi:10.1007/bf02764723.
- [21] A. Vera López and J. Vera López, Classification of finite groups according to the number of conjugacy classes. II, *Israel J. Math.* **56** (1986), 188–221, doi:10.1007/bf02766124.
- [22] C. W. Wynn, *Supercharacter theories of Camina pairs*, Ph.D. thesis, Kent State University, 2017, <https://www.proquest.com/docview/1896119212>.