

# Enumeration of Optimal Equidistant Codes

Galina Bogdanova <sup>1,\*</sup>  and Todor Todorov <sup>1,2</sup>

<sup>1</sup> Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria; t.todorov@ts.uni-vt.bg

<sup>2</sup> Faculty of Mathematics and Informatics, St. Cyril and St. Methodius University of Veliko Tarnovo, 5003 Veliko Tarnovo, Bulgaria

\* Correspondence: galina@math.bas.bg

**Abstract:** Problems of search and enumeration of binary and ternary equidistant codes are considered in the paper. We investigate some combinatorial algorithms and develop specialized computer packages to find non-equivalent optimal binary and ternary equidistant codes for  $3 \leq d \leq n \leq 9$ .

**Keywords:** optimal codes; equidistant codes; enumeration of codes

## 1. Introduction

We can define a  $q$ -ary  $(n, M, d)_q$  code as a set of  $M$  vectors with length  $n$  over the alphabet  $Z_q = \{0, 1, 2, \dots, q - 1\}$  and with additional conditions that any two different vectors (codewords) differ in at least  $d$  coordinates. If we add a restriction for all the codewords to have Hamming weight  $w$ , then it becomes a constant weight. The notation is  $(n, M, d, w)_q$ . Research on such codes is presented in [1–8]. If  $A_q(n, d, w)$  indicates the biggest value  $M$  for an  $(n, M, d, w)_q$  code, then a code with parameters  $(n, A_q(n, d, w), d, w)_q$  is considered optimal. More results about such codes can be found in [2,5,7,8]. An  $(n, M, d)_q$  equidistant code is a set of  $M$  codewords of length  $n$  over the alphabet  $\{0, 1, \dots, q - 1\}$ , and any two different codewords differ in exactly  $d$  positions. We can define an equidistant constant weight code  $(n, M, d, w)_q$  as a set of  $M$  codewords of length  $n$  over the alphabet  $\{0, 1, \dots, q - 1\}$ , any two different codewords differ in exactly  $d$  positions and all its codewords have the same weight  $w$ .

Let  $B_q(n, d)$  be the maximum number of codewords in an equidistant code with length  $n$  and distance  $d$  between codewords. Let  $B_q(n, d, w)$  be the maximum number of codewords in an equidistant constant weight code with length  $n$ , distance  $d$  between codewords and weight  $w$  of each codeword. An equidistant code with  $B_q(n, d)$  codewords is called optimal. An equidistant constant weight with  $B_q(n, d, w)$  codewords is called optimal. Equidistant codes have been investigated in [9–13].

In this paper, classification of optimal binary and ternary equidistant codes for  $3 \leq d \leq n \leq 9$  is considered. There are two important problems in coding theory related to code generation and code enumeration up to equivalence. These are  $NP$ -complete tasks and that is why they are not trivial [14,15].

Section 2 considers some general bounds related to constant weight codes and equidistant codes. Section 3 presents computer methods that are used for a computer search of optimal equidistant codes. Section 4 gives the enumeration of binary and ternary equidistant codes by computer methods.

## 2. Preliminaries

The following theorems consider some known restrictions and bounds that will be used for code searching and enumeration. Let  $A_q(n, d)$  denote the maximum number of codewords of a code over the alphabet with  $q$  elements, distance  $d$  and length  $n$ .



**Citation:** Bogdanova, G.; Todorov, T. Enumeration of Optimal Equidistant Codes. *Mathematics* **2022**, *10*, 740. <https://doi.org/10.3390/math10050740>

Academic Editors: Jiyou Li and Victor Mitrana

Received: 23 December 2021

Accepted: 24 February 2022

Published: 26 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Theorem 1.**

- a.  $A_q(n, d, n) = A_{q-1}(n, d).$
- b.  $A_q(n, 2, w) = \binom{n}{w} (q - 1)^{w-1}.$
- c. *If  $A_q(n, d, w) \geq 2$ , then  $d \leq 2w$  and  $d \leq n$ .*
- d.  $A_q(n, 2w, w) = \lfloor n/w \rfloor.$
- e. *(Johnson bound)  $A_q(n, d, w) \leq \frac{n}{n-w} A_q(n - 1, d, w).$*
- f.  $A_q(n, d, w) \leq \frac{n(q-1)}{w} A_q(n - 1, d, w - 1).$

**Theorem 2.**

- a.  $A_3(n, d, w) \leq A_2(n, 2(d - w), w)$  if  $\frac{d}{2} \leq w < d.$
- b.  $A_3(n, d, w) \geq 2$  exactly when  $d \leq \min\{2w, n\}.$
- c.  $A_3(n, d, w) \geq 3$  exactly when  $d \leq \min\{2w, \frac{n}{3} + w, \frac{5n}{3} - w\}.$
- d. [8]  $A_3(n, d, w) = 1$  exactly when  $d > \min\{2w, n\}.$
- e.  $A_3(n, d, w) = 2$  exactly when  $\min\{n/3 + w, 5n/3 - w\} < d \leq \min\{2w, n\}.$

**Theorem 3.**

- a.  $B_q(n, n) = q.$
- b. [16]  $B_q(n, d) = 1 + B_q(n, d, d).$
- c. (Plotkin) [12,17]  $B_q(n, d) \leq \frac{dq}{dq - n(q-1)}$  if the denominator is positive.
- d. (Delsarte) [18]  $B_q(n, d) \leq (q - 1)n + 1.$

These theorems are extensively used in computer algorithms used for code generation and code enumeration. Bounds for the size of codes restrict searching space and reduce the calculation time.

Moreover, in Section 4 will be discussed how code generation and code enumeration can be performed concurrently to optimize the performance of the whole process.

**3. Code Searching**

In [5,6,8,9], some combinatorial methods for obtaining the bounds on the size of the codes are presented. Here we will present methods for a computer search of binary and ternary equidistant codes that we will use in the enumeration process.

We say that the lexicographic order of two columns of a code  $b = \{b_1, b_2, \dots, b_M\}$  and  $c = \{c_1, c_2, \dots, c_M\}$  is preserved if  $b_i = c_i, i = 1, \dots, k$  and  $b_{k+1} < c_{k+1}$ , for some  $k, k < M$ .

*3.1. Backtrack Search*

- Add the all-zero codeword to the code.
- Fix the second codeword of the searched code with  $a = \underbrace{0, 0, \dots, 0}_{n-d} \underbrace{1, 1, \dots, 1}_d.$
- The set of codewords that will be used for searching contains all the codewords that are at a distance exactly equal to  $d$  from the codeword  $a$ . They are stored lexicographically.
- Backtrack search—a codeword from the considered space becomes part of the code if the distance stays equal to  $d$  and the lexicographic order of the columns of the code is preserved.
- When we are at the end of the search space, we check the size of the generated code to find if its size is bigger than what was previously found. The search continues by changing the codewords backward on the search tree.

### 3.2. Lexicographic Search

- Select the set of fixed codewords.
- The set of codewords that will be used for searching contains all the codewords that are at a distance exactly equal to  $d$  from the fixed codewords. They are stored lexicographically.
- A codeword from the considered space becomes part of the code if the distance stays exactly equal to  $d$  and the lexicographic order of the columns of the code is preserved.
- As an opposite to the previous method there is no step back. At the end of the search space we have the final generated code.

There are several variations:

- Automatically change the starting set of codewords to find the code with the larger size.
- Cyclically shift the search set that gives better results in some cases rather than the standard order.

### 3.3. Graph Clique Search

Searching for better values of  $A_q(n, d)$  can be considered a graph clique search problem.

Let us consider a graph with vertices related to a codeword with a length  $n$  and a weight  $d$  and a set of edges that connects a pair of vertices only if the distance between the codewords in these vertices is exactly equal to  $d$ . Then we can apply an algorithm for the maximum clique searching in such a graph.

A more detailed explanation of this approach, in general, can be found in [19].

### 3.4. Construction with Extension

All search methods, described up to now, work only for small parameters or only in some special cases. That is why we combine them with extension methods. The essence is that the construction of the  $(n, M, d)_q$  code can be performed by extension of an already known  $(n - 1, M', d)_q$  code. We can use this fact and try to extend all non-equivalent  $(n - 1, M', d)_q$  codes in order to find the optimal one.

## 4. Enumeration of Binary and Ternary Equidistant Codes

**Definition 1.** Let's define the equivalence between two codes as a set of transformations that translates one of the codes into the other. These transformations are exchange of coordinates in the code or permutation of symbols.

In Propositions 1, 2, 3 and Theorem 4 we present and proof some new combinatorial results related to enumeration of equidistant codes.

**Proposition 1.** There exists an optimal equidistant code with parameters  $(n, q, n)_q$ ,  $n \geq 2$ . This code is unique up to equivalence.

It can be derived from Theorem 3a.

**Theorem 4.** When  $3 \leq q < 9$  and  $n > 3$  then  $B_q(n, 3) = 9$  and when  $q \geq 9$  and  $n > 3$  then  $B_q(n, 3) = q$ .

**Proof.** We may assume that all the codewords are ordered in a lexicographic way.  $\square$

1. Without loss of generality  $00 \dots 00000$  and  $00 \dots 00111$  are from the code.
2. If there are only zeros in the first  $n - 3$  coordinates, then each of the remaining codewords is of type  $0000ttt$ , where  $t$  is a random element from the alphabet. It is evident that  $B_q(n, 3) \leq q$ .
3. If in any of the first  $n - 3$  coordinates there is a non-zero element, then without loss of generality  $00 \dots 01012$  is a codeword.

- If there are only zeros in the first  $n - 4$  coordinates, then the code is  $(4, M, 3)_q$ , for which it is well known that  $M \leq 9$  [9].
- If  $n \geq q + 1$  and in any of the first  $n - 4$  coordinates there is a non-zero element, then without loss of generality  $00 \dots 10013$  is from the code. We can see easily that a word with distance 3 from the already known four codewords is equivalent to  $00 \dots 100014$  and so on until we obtain the  $(q + 1, q, 3)_q$  code.

For instance, the obtained  $(10, 9, 3)_9$  code for  $q = 9$  is:

$$\begin{array}{r}
 00 \dots 00000 \\
 00 \dots 00111 \\
 00 \dots 01012 \\
 00 \dots 10013 \\
 \dots \\
 10 \dots 00018
 \end{array} \tag{1}$$

That is why if  $3 \leq q < 9$  and  $n > 3$ , then  $B_q(n, 3) = 9$  and when  $q \geq 9$  and  $n > 3$ , then  $B_q(n, 3) = q$ .

**Construction 1.** The  $(n + k, M, d)_q$  code can be derived from  $(n, M, d)_q$  code  $A: \{(\underbrace{0 \dots 0}_k, a) \mid a \in A\}$ .

**Proposition 2.** Equidistant codes with parameters  $(n, 9, 3)_q$  for  $n \geq 4$  and  $q < 9$  exist and are optimal. Each of the codes of this family is unique up to equivalence.

It follows from Theorem 4 and Construction 1.

**Proposition 3.** Equidistant codes with parameters  $(n, q, 3)_q$  for  $n \geq q + 1$  and  $q \geq 9$  exist and are optimal. There are two families of such codes.

This follows from Theorem 4 and Construction 1.

There is a direct connection between equidistant codes and constant-weight codes. Let  $C$  be an  $(n, M, d, w)_q$  equidistant constant-weight code for  $w = d$  and  $C_0 = C \cup \{0\}$  be an  $(n, M_0, d)_q$  equidistant code. This relation is expressed by Theorem 3b. Also we can use code lengthening approach to construct an  $(n, M', d, d)_q$  code  $C$  from  $(n - 1, M', d, d)_q$  code  $C'$ .

The next theorem follows from Theorem 1e, Theorem 1f and Theorem 3b.

**Theorem 5.** Any  $(n, M, d)_q$  equidistant code  $C$  contains  $(n - 1, M', d)_q$  codes with  $M' = \lceil (M - 1) \frac{n-d}{n} \rceil + 1$  codewords.

There are two main problems: the existence and classification of equidistant codes.

The first problem is to construct all the  $(n, M, d, w = d)_q$  codes with  $M$  codewords, which contain  $C'$  as a sub-code. The search space will only be the vectors which are at a distance exactly  $d$  from code  $C'$ . We will only have to care about the distance between codewords. This is a maximum clique problem, and we use a backtrack search. In the  $q$ -ary case we can construct the graph whose vertices represent  $q$ -ary vectors of length  $n$ . We join three vertices by an edge if, and only if, the Hamming distance between the vectors is exactly  $d$ . Then what we are interested in is the value of  $B_q(n, d) = B_q(n, d, d) + 1$ , the size of the largest clique in this graph.

The second problem is to find all the inequivalent codes. Let  $C_1$  and  $C_2$  are ternary equidistant codes. Two codes  $C_1$  and  $C_2$  are equivalent if  $C_2$  can be obtained from  $C_1$  by: a permutation of the coordinates of the code, a permutation of the alphabet symbols in one or more coordinates, or any combination thereof. If we wish to compare two codes, we first convert those codes into graphs, and then check the graphs for isomorphism.

To obtain the classification results for  $q$ -ary equidistant codes, we use some theoretical and software tools [20].

Tables 1–3 present the main new classification results for equidistant codes. For code generation, we use computer algorithms described in Section 3 that have proved their correctness in many similar cases. For example, using the same techniques in [9] we found the exact values for  $B_q(n, d)$ , for many different values of  $q, n$  and  $d$  and all these results are also confirmed by combinatorial methods.

**Table 1.** Classification of binary equidistant codes.

$n$	$d = 2$		$n$	$d = 4$		$n$	$d = 6$		$n$	$d = 8$		$n$	$d = 10$	
	$M$	#		$M$	#		$M$	#		$M$	#		$M$	#
2	2	1	4	2	1	6–8	2	1	8–11	2	1	10–14	2	1
3	4	1	5	2	2	9	4	1	12	4	1	15	4	1
4	4	2	6	4	1	10	6	1	13	4	2			
5–15	$n$	1	7	8	1	11	12	1	14	8	4			
			8–15	8	3	12	12	8	15	16	5			
						13–14	13	1						
						15	14	2						

**Table 2.** Classification of ternary equidistant codes.

$n$	$d = 3$	$d = 4$	$d = 5$	$d = 6$	$d = 7$	$d = 8$	$d = 9$	$d = 10$
	$M/\#$	$M/\#$	$M/\#$	$M/\#$	$M/\#$	$M/\#$	$M/\#$	$M/\#$
4	9/1	3/1						
5	9/1	6/1	3/1					
6	9/1	7/1	4/1	3/1				
7	9/1	8/1	7/2	3/2	3/1			
8	9/1	8/3	8/3	9/9	3/1	3/1		
9	9/1	8/3	8/3	12/10	6/6	3/2	3/1	
10	9/1	8/3	8/3	15/1	10/1	6/3	3/2	3/1

The symbol # denotes the number of inequivalent equidistant codes. The exact values for  $d = 3$  are obtained by Theorem 4.

**Table 3.**  $(n, M, d)_3$  inequivalent equidistant codes.

$n$	#	$(n, M, d)_3$ Codes	
		$(n, M, 4)_3$ codes	$(n, M, 5)_3$ codes
5	1	(00000, 01111, 10122, 12201, 21220, 22012)	
6	1	(000000, 001111, 010122, 012201, 100221, 102102, 220101)	
7	1	(0000000, 0001111, 110011, 0111100, 1010101, 1011010, 1100110, 1101001)	
8	3	(1) (00000000, 00001111, 00110011, 00111100, 01010101, 01011010, 01100110, 01101001) (2) (00000000, 00001111, 00110011, 00111100, 01010101, 01011010, 01100110, 10010110) (3) (00000000, 00001111, 00110011, 00111100, 01010101, 01011010, 10010110, 10011001)	
9–10	3	The codes are obtained by Construction 1 up to equivalence.	
		$(n, M, 5)_3$ codes	
6	1	(000000, 011111, 101222, 222012)	
7	2	(1) (0000000, 0011111, 0101222, 0222012, 1012202, 1102110, 1121001) (2) (0000000, 0011111, 0101222, 0222012, 1012202, 1200211, 1211020)	

**Table 3.** *Cont.*

$n$	#	$(n, M, d)_3$ Codes
8	3	(1) (00000000, 00011111, 00101222, 00222012, 01012202, 01102110, 01121001, 11001012) (2) (00000000, 00011111, 00101222, 00222012, 01012202, 01102110, 01121001, 21001012) (3) (00000000, 00011111, 00101222, 00222012, 01012202, 01102110, 01211020, 11001012)
9–10	3	The codes are obtained by Construction 1 up to equivalence.
$(n, M, 6)_3$ codes		
7	2	(1) (0000000, 0111111, 0222222) (2) (0000000, 0111111, 1012222)
8	9	(00000000, 00111111, 00222222, 11001122, 11112200, 11220011, 22002211, 22110022, 22221100) —the first of these codes.
9	10	(000000000, 000111111, 000222222, 011001122, 011112200, 011220011, 101010212, 101121020, 101202101, 110012021, 110120102, 110201210) —the first of these codes.
10	1	(0000000000, 0000111111, 0000222222, 0011001122, 0022002211, 0101010212, 0120201012, 0202101202, 0210021210, 1001201210, 1020011202, 1200002112, 2002021012, 2010100212, 2100001221)
$(n, M, 7)_3$ codes		
8	1	(00000000, 01111111, 02222222)
9	6	(1) (000000000, 001111111, 010122222, 112200112, 121212020, 122021201) (2) (000000000, 001111111, 010122222, 112200112, 121212020, 222011202) (3) (000000000, 001111111, 010122222, 112200112, 121212020, 222022101) (4) (000000000, 001111111, 010122222, 112200112, 121222001, 122011220) (5) (000000000, 001111111, 010122222, 112200112, 121222001, 222001221) (6) (000000000, 001111111, 010122222, 112200112, 121222001, 222012120)
10	1	(0000000000, 0001111111, 0010122222, 0112200112, 0121212020, 1100211202, 1102120021, 1201002122, 2120102101, 2211110002)
$(n, M, 8)_3$ codes		
9	2	(1) (000000000, 011111111, 022222222) (2) (000000000, 011111111, 101222222)
10	3	(1) (0000000000, 0011111111, 1100112222, 1122220011, 2211222200, 2222001122) (2) (0000000000, 0011111111, 1100112222, 1122220011, 2212021202, 2221202120) (3) (0000000000, 0011111111, 1100112222, 1212220012, 2122021201, 2221202120)
$(n, M, 9)_3$ codes		
10	2	(1) (0000000000, 0111111111, 0222222222) (2) (0000000000, 0111111111, 1012222222)

All the results related to code enumeration are obtained using the combinatorial restrictions from Proposition 1, 2, 3 and Theorem 4 and proved computer methods for code enumerations similar to those described in [21]. The main idea of the algorithm is to transform the problem of code equivalence to the problem of isomorphism of binary matrices. Finally, in Table 3 are listed all non-equivalent codes found by computer algorithms and combinatorial considerations so their correctness could be easily checked.

**Proposition 4.** *There are # inequivalent equidistant codes with parameters  $(n, M, d)_3$  for  $5 \leq n \leq 10$ . The codewords of the  $(n, M, d)_3$  codes are presented in Table 3 (up to equivalence).*

The exact values for  $n = d$  are obtained by Theorem 3a.

## 5. Conclusions

In the paper, we use a specialized computer package QPlus to enumerate optimal binary and ternary equidistant codes for some parameters. Combinatorial methods are implemented in computer algorithms to restrict the number of codewords in the search sets.

The developed approach and techniques, related to the presented combinatorial objects and their application for digital data protection, are implemented in the project “Digital Accessibility for People with Special Needs: Methodology, Conceptual Models and Innovative Ecosystems”. The research in the project focuses mainly on semantic models for presenting unstructured data in the field of digital ecosystems and building an ontology for accessibility of people with disabilities. Some models for effective data organization and appropriate methods for creating, analysing, and protecting innovative accessible digital ecosystems are also explored. The research project develops methods for solving security problems by using discrete structures to ensure data security. Some of the codes considered in the paper are used for that protection.

**Author Contributions:** Data curation, T.T.; Investigation, G.B.; Methodology, G.B.; Resources, T.T.; Software, G.B.; Validation, T.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research work has been partially supported by the National Science Fund of Bulgaria (scientific project “Digital Accessibility for People with Special Needs: Methodology, Conceptual Models and Innovative Ecosystems”), Grant Number KP-06-N42/4, 8 December 2020.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Agrell, E.; Vardy, A.; Zeger, K. Upper bounds for constant-weight codes. *IEEE Trans. Inform. Theory* **2000**, *46*, 2373–2395.
2. Brouwer, A.; Shearer, J.; Sloane, N.; Smith, W. A new table of constant weight codes. *IEEE Trans. Inform. Theory* **1990**, *36*, 1334–1380. [[CrossRef](#)]
3. Chee, Y.; Ling, S. Constructions for  $q$ -ary constant-weight codes. *IEEE Trans. Inform. Theory* **2007**, *53*, 135–146. [[CrossRef](#)]
4. Dai, P.; Wang, J.; Yin, J. Two series of equitable symbol weight codes meeting the Plotkin bound. *Des. Codes Cryptogr.* **2015**, *74*, 15–29. [[CrossRef](#)]
5. Gashkov, I.; Taub, D. New Optimal Constant Weight Codes. *Electron. J. Comb.* **2007**, *14*, N13. [[CrossRef](#)]
6. Ge, G.; Wu, D. Some new optimal quaternary constant weight codes. *Sci. China Ser. F.* **2005**, *48*, 192–200. [[CrossRef](#)]
7. Krotov, D.S. On Diameter Perfect Constant-Weight Ternary Codes. *Discret. Math.* **2008**, *308*, 3104–3114. [[CrossRef](#)]
8. Östergård, P.; Svanström, M. Ternary constant weight codes. *Electron. J. Comb.* **2002**, *9*, R41. [[CrossRef](#)]
9. Bogdanova, G.; Todorov, T.; Zinoviev, V. On construction of  $q$ -ary equidistant codes. *Probl. Inf. Transm.* **2007**, *43*, 13–36. [[CrossRef](#)]
10. Boyvalenkov, P.; Delchev, K.; Zinoviev, D.V.; Zinoviev, V.A. On  $q$ -ary Codes with Two Distances  $d$  and  $d + 1$ . *Probl. Inf. Transm.* **2020**, *56*, 33–44. [[CrossRef](#)]
11. Kathuria, A.; Arora, S.; Batra, S. On traceability property of equidistant codes. *Discret. Math.* **2017**, *340*, 713–721. [[CrossRef](#)]
12. van Lint, J.H. A theorem on equidistant codes. *Discret. Math.* **1973**, *67*, 353–358. [[CrossRef](#)]
13. Semakov, N.V.; Zinoviev, V.A. Equidistant  $q$ -ary codes with maximal distance and resolvable balanced incomplete block designs. *Probl. Peredachi Inf.* **1968**, *4*, 3–10.
14. Kaski, P.; Östergård, P. *Classification Algorithms for Codes and Designs*; Springer: Berlin, Germany, 2006.
15. Knuth, D. *Art of Computer Programming*; Addison-Wesley Professional: Boston, MA, USA, 1998.
16. Fu, F.; Klove, T.; Luo, Y.; Wei, V. On equidistant Constant Weight codes. In Proceedings of the WCC’2001 Workshop on Coding and Cryptography, Paris, France, 8–10 January 2001; pp. 225–232.
17. Plotkin, M. Binary codes with specified minimum distance. *IRE Trans. Inform. Theory* **1960**, *6*, 445–450. [[CrossRef](#)]
18. Delsarte, P. Bounds for unrestricted codes by linear programming. *Philips Res. Rep.* **1972**, *27*, 47–64.

19. Östergård, P. A fast algorithm for the maximum clique problem. *Discret. Appl. Math.* **2002**, *120*, 195–205. [[CrossRef](#)]
20. Bogdanova, G.; Todorov, T.; Todorov, V. QPlus—Computer package for coding theory research and education. *Int. J. Comput. Math.* **2011**, *88*, 443–451. [[CrossRef](#)]
21. Bouyukliev, I. About the code equivalence. *Adv. Coding Theory Cryptol.* **2007**, *3*, 126–151.