Combinatorial optimization algorithms for the design of codes: a survey

Roberto Montemanni \(^{1,2,*}\)

\(^{1}\) Dalle Molle Institute for Artificial Intelligence, Lugano, Switzerland
\(^{2}\) University of Applied Sciences of Southern Switzerland, Lugano, Switzerland

Received 04 December 2014
Accepted 15 January 2015

Abstract—Code design problems are central in information theory and have applications in different fields ranging from telecommunications to bioinformatics. In this paper we survey the results achieved in the last decades for different types of codes, covering constant weight binary codes, quaternary codes and permutation codes. The focus of the survey is mainly on approaches based on combinatorial optimization and graph theory that have gained popularity in the last decade and are likely to play a central role in the codes design research in the forthcoming years, possibly in a fashion where the new ideas are hybridized with the older contributions.

Published 01 March 2015
Copyright © ORLab Analytics Inc. All rights reserved.

Introduction

Code design has a central role in information theory with applications in many fields, and in particular in telecommunications. A code is a set of words of a given length, composed from a given alphabet, and with some application-dependent characteristics that typically guarantee some form of separation or orthogonality between the words. Codes are usually adopted for correcting errors in messages, or for modulation of signals in communications. Codes have also found use in some biological applications more recently. The target is usually to have codes with as many words as possible, according to some constraints. In general, a code is a set of words of a given length defined over a given (application dependent) alphabet that fulfills some given design properties.

The most typical constraint is on the Hamming distance between each pair of words. The Hamming distance \(d(x, y)\) between two words \(x\) and \(y\) is defined as the number of positions in which they differ. The minimum distance of a code is the minimum Hamming distance between any pair of words of the code. Other side-constraints that depend on the specific application, for which codes are defined, are also present. In the engineering applications, maximizing the number of words means efficiency and is important because this provides engineers with the widest possible freedom when implementing communication systems or other specific applications. Depending on the underlying real applications, several types of code are of practical interest. The maximization of the number of words however, independently of the characteristics of the code, makes it natural to formalize the problem as a combinatorial optimization problem.

Many approaches to solve these code design problems have been proposed in recent decades. So far, most research effort to construct good codes has been based on abstract algebra and group theory, while only a marginal exploration of combinatorial optimization (and graph theoretical) concepts has been carried out only recently, notwithstanding the intrinsic optimization component of the problem mentioned before. The optimization algorithms developed so far, although effective, are fairly detached from the previous contributions that were based on more abstract mathematics-based tools.

The aim of the present paper is to describe and discuss some most relevant approaches based on combinatorial optimization and graph theory currently available in the literature for different code design problems. In particular, constant weight binary codes, quaternary codes and permutation codes will be covered. The reader should however be aware that many other possible codes, justified by different applications, exist. A classification of the different codes can be found in Cover & Thomas (1991) and Ryan & Lin (2009). The objective of the problem is to find a code that fulfills all the constraints and contains the maximum possible number of words.

\(*\) Correspondence: Roberto Montemanni, Dalle Molle Institute for Artificial Intelligence, University of Applied Sciences of Southern Switzerland, Galleria 2, 6928 Manno, Canton Ticino, Switzerland
E-mail: roberto@idsia.ch
Constant weight binary codes

A constant weight binary code is a set of binary vectors of length \( n \), weight \( w \) and minimum Hamming distance \( d \). The weight of a binary vector (or word) is the number of 1’s in the vector. The minimum distance of a code is the minimum Hamming distance between any pair of words. The maximum possible number of words in a constant weight code is referred to as \( A(n, d, w) \). An example of constant weight binary code is provided in Figure 1.

```
110011000010 100110001011
011000010110 110001011001
001100001011 000101100110
010110011001 000010110011
101100110000 011001100001
```

Fig. 1. An optimal (12, 6, 5) constant weight binary code

The problem of determining \( A(n, d, w) \) is equivalent to the packing problem: determine the maximum number \( D(t, k, v) \) of \( k \)-subsets of a \( v \)-set such that no \( t \)-subset is covered more than once (Brouwer, 1995; Mills & Mullin, 1992). Apart from their important role in the theory of error-correcting codes (MacWilliams & Sloane, 1977), constant weight codes have also found application in fields as diverse as the design of demultiplexers for nano-scale memories (Kuekes et al., 2006), the construction of frequency hopping lists for use in GSM networks (Moon et al., 2005) and the design of DNA codes for use in DNA barcoding and DNA computing (King, 2003). Accounts of the theory of constant weight codes can be found in MacWilliams & Sloane (1977) and Brouwer et al. (1990).

A detailed account of upper bounds for \( A(n, d, w) \) can be found in Agrell et al. (2000). Lower bounds for \( A(n, d, w) \) are usually obtained constructively, either via theoretical mathematical constructions, or via results from permutation groups. Code constructions for \( n \leq 28 \) can be found in Brouwer et al. (1990). In Smith et al. (2006) a comprehensive set of constructions was described for the parameter sets appropriate to the frequency hopping application. These parameter sets were \( 29 \leq n \leq 63 \) and \( 5 \leq w \leq 8 \) with \( d = 2w - 2 \), \( d = 2w - 4 \) and \( d = 2w - 6 \). A small number of improvements to the values in Smith et al. (2006) can be found in Gashkov & Taub (2007), Gashkov et al. (2007), Stinson et al. (2007) and (derived from a group divisible design) in Ge (2007). Heuristic constructions of constant weight codes (which do not attempt to identify or store any mathematical structure) are useful in applications. For longer codes they often give the best-known code, particularly when no good mathematical structure has been identified. However, these mechanisms must be well designed if they are to be effective. Simply applying massive computer power without a good algorithm is unlikely to lead to a good code. Concerning metaheuristic algorithms, both simulated annealing (Gamal et al., 1987) and tabu search (Bland & Bayliss, 1997) have been used to construct constant weight codes, although the number of results presented in these papers is limited. Some metaheuristic methods, based on lexicographic search, clique search and algorithms derived from these two main ideas, have been proposed in Montemanni & Smith (2009a), leading to new state-of-the-art results for many parameter settings. The systematic application of group theory constructions hybridized with heuristic algorithms has been discussed in Smith & Montemanni (2012b), leading to further improvements of best-known results. In Neelakandan (2010) some methods based on a primitive combination of mathematical programming techniques with some greedy heuristics, has been discussed. In conclusion, it can be noted from the results in Smith et al. (2006) that when no good mathematical construction is available and methods from Brouwer et al. (1990) using permutation groups cease to be feasible in a reasonable run time, then heuristic approaches become the default method.

Quaternary codes

Quaternary codes are sets of words of fixed length \( n \) over the alphabet \( \{A, C, G, T\} \). The words of a code have to satisfy the following combinatorial constraints. For each pair of words, the Hamming distance has to be at least \( d \) (constraint HD); a fixed number (here taken as \( \lceil n/2 \rceil \)) of letters of each word have to be either G or C (constraint GC); the Hamming distance between each word and the Watson-Crick complement (or reverse-complement) of each word has to be at least \( d \)
DNA codes have applications to information storage and retrieval in synthetic DNA strands. They are used in DNA computing (Adleman, 1994; Arita & Kobayashi, 2002; Bi et al, 2003; Marathe et al, 2001), as probes in DNA microarray technologies (Fodor et al, 1991; Shena et al, 1995) and as molecular bar codes for chemical libraries (Brenner, 1997; Brenner & Lerner, 1992; Shoemaker et al, 1996). In these applications the primary property required is that short single strands of DNA (oligonucleotides) should hybridize with their Watson-Crick complements, but other undesirable hybridizations should be unlikely (Chen et al, 2003). Thus two oligonucleotides in the library should be dissimilar and an oligonucleotide in the library should be dissimilar to the Watson-Crick complement of any oligonucleotide in the library. This is achieved by specifying the minimum Hamming distance (minimum number of differences) between words, and between words and the reverse complements of words. Such constraints have been used in Frutos et al (1997), Gaborit & King (2005), King (2003), Li et al (2002) and Marathe et al (2001), where further references can be found. Another desirable property concerns the melting temperature. DNA melting is the process by which double-stranded DNA unwinds and separates into single-stranded strands through the breaking of hydrogen bonding between the bases. Similar melting temperatures can be approximately achieved by ensuring that each word contains the same number of positions which are either G or C (giving constant G-C content), see Frutos et al (1997). This allows hybridization of multiple words to take place simultaneously (Shoemaker et al, 1996). More accurate models of melting temperature are available in Marathe et al (2001).

Upper bounds for DNA codes are presented in Gaborit & King (2005) and in Marathe et al (2001) and constructive lower bounds using coding theory, stochastic search, a template-map strategy, genetic algorithms and lexicographic codes have been proposed (Chee & Ling, 2008; Deaton et al, 1996; Frutos et al, 1997; Gaborit & King, 2005; King, 2003; Tulpan & Hoos, 2003; Tulpan et al, 2002; Montemanni et al, 2014c; Koul, 2010 and Aboluion, 2011).

Metaheuristic algorithms mainly based on lexicographic search and clique search evolutions, combined in a variable neighborhood search framework, have been presented in Montemanni & Smith (2008). A simulated annealing method was proposed in Montemanni & Smith (2009b), while an evolutionary algorithm was discussed in Koul (2010) and Montemanni et al (2014c). A range of metaheuristic methods and mathematical constructions has been applied to other DNA codes design problem in Tulpan et al (2014), without however any form of hybridization. When a coding theory construction is available, it may compare very favorably with the best algorithmic construction. However, there are many cases where the best-known result is an algorithmic construction. Algorithmic constructions have the advantage that they can be easy to modify if additional constraints need to be incorporated. Additional constraints can be designed to give insensitivity to frame-shifts, avoid secondary structure, or to more accurately model melting temperature, see Brenner & Lerner (1992) and Milenkovic and Kashyap (2006). When a coding theory construction is used, the elements mentioned above can be included via post-processing, but the benefits of the theoretical construction may then be lost. In applications in which codewords (possibly of variable length) concatenate, other constraints arise, which have been handled with coding theory (e.g. Arita & Kobayashi, 2002; Kobayashi et al, 2003) or studied using formal languages (e.g. Kari et al, 2003).

**Permutation codes**

A permutation code is a set of permutations in the symmetric group $S_n$ of all permutations on $n$ elements. The words are the permutations and the code length is $n$. The ability of a permutation code to correct errors is related to the minimum Hamming distance of the code. The minimum Hamming distance $d$ is then the minimum distance taken over all pairs of distinct permutations. The maximum number of words in a code of length $n$ with minimum distance $d$ is denoted by $M(n, d)$. An example of permutation code is provided in Figure 3.

<table>
<thead>
<tr>
<th>Code</th>
<th>Hamming Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTTC GGTT</td>
<td></td>
</tr>
<tr>
<td>GTCA AGGA</td>
<td></td>
</tr>
<tr>
<td>ACTG TTGG</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2. An optimal (4, 3) quaternary code fulfilling constraints HD, GC and RC
Permutation codes (sometimes called permutation arrays) have been proposed in Han Vinck (2000) for use with a specific modulation scheme. An account of the rationale for the choice of permutation codes can be found in Chu et al. (2004) (see also Huczynska, 2006 and Pavlidou et al., 2003). Permutations are used to ensure that power output remains as constant as possible. As well as white Gaussian noise the codes must combat permanent narrow band noise from electrical equipment or magnetic fields, and impulsive noise.

A central question in the theory of permutation codes is the determination of $M(n, d)$, or of good lower bounds for $M(n, d)$. The most complete contribution to this question is in Chu et al. (2004). Codes derived both from permutation groups, heuristic algorithms, and a basic combination of the two, have been discussed in Smith & Montemanni (2012a). The results of the application of some metaheuristic methods to the problem had been earlier detailed in Montemanni et al. (2012). Very recently, some preliminary results of novel approaches have been discussed in Barta et al. (2014). A branch and bound approach based on graph theoretical concepts has been finally presented in Montemanni et al. (2014a) and Montemanni et al. (2014b). New lower bounds were provided by these methods. More recently, different methods, both based on permutation groups and heuristic algorithms, have been presented in Smith & Montemanni (2012a). Not so long ago, a new paper based on the use of maximum clique programs over graphs generated by group theoretical analysis has been presented in Janiszczak et al. (2014).

Conclusion

Code design problems have been studied for decades, with very important developments and results achieved. Codes play a central role in telecommunication, since many real applications are based on different types of error-correcting codes, where the characteristics of the codes themselves depend on the underlying application. The increasing complexity of communication infrastructures, due to media saturation and to the use of more unstable channels, is leading to the need of larger and more robust codes to keep and improve the communication quality. This brings the research to unexplored regions, where current tools from abstract mathematics are likely to be less and less successful, due to the increasing difficulties in capturing the low-level characteristics and the more complex structure of the larger codes into abstract mathematical models. Here the use of combinatorial optimization-based tools is going to play a central role for the intrinsic complexity of the design problems mentioned before. The present paper is meant to address combinatorial optimization researchers in this direction, providing a reasoned survey of the literature currently available.

References


