IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 January 2024, accepted 16 January 2024, date of publication 19 January 2024,
date of current version 25 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3356354

RESEARCH ARTICLE

Greedy Iterative and Meta-Heuristic Clustering With Coded Caching and Slepian-Wolf Compression for Correlated Content

BENJAMIN ROSEN AND LING CHENG, (Senior Member, IEEE)
School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg 2000, South Africa

Corresponding author: Ling Cheng (ling.cheng@wits.ac.za)

This work was supported in part by the National Research Foundation of South Africa under Grant 148765, Grant 132651, and Grant 129311.

ABSTRACT Content caching has emerged as an effective approach to combat the increasing strains on our current network infrastructure. This method is further improved when combining caching with source coding. However, additional complexity is incurred by creating this hybrid method, as the source coding component comes with associated feasibility constraints and decoding costs. This paper presents an approach to balance this complexity with the coding gains by selecting the best-performing subset of files to compress, while the others are left uncoded. This problem is shown to be NP-hard in general and difficult to solve in an iteration-free manner. To this end, two novel approaches are outlined: an iterative-based solution, which uses the features of the entropy function to derive the most suitable files to compress jointly, and a meta-heuristic version, which is based on the Genetic Algorithm. When compared to an exhaustive search, the proposed solutions are found to be sub-optimal but falling above the 90th percentile of all possible solutions on average. Significantly, the iterative method produces results within one percentile of the meta-heuristic approach yet it finds a solution 2.31 times faster. The iterative approach has an additional benefit, in that it is able to predict the relative gains when adding more files to a compression group. It is thus able to terminate prematurely if the estimated gains are less than a chosen threshold.

INDEX TERMS Slepian-wolf coding, distributed source coding, coded caching, iterative optimisation.

I. INTRODUCTION

Recently, there has been a relentless increase in the amount of data traffic as the number of Internet users and Internet-connected devices grows. This, together with ever-improving Internet speeds and availability, has put immense strain on the current network infrastructure. Thus, current research is focused on intelligent methods to organise and deliver content without relying on ad hoc network usage. One promising method is content caching [1], where relevant information is downloaded to user devices during off-peak hours based on expected demand. The server is then able to provide the content to consumers at a reduced rate when the network is constrained.

In practical applications, the files being requested are often correlated to one another, since many types of content, such as current news and popular videos, have a high degree of similarity [2]. Consequently, new techniques have been proposed to improve the efficacy of content caching by taking the correlation between files into consideration [3], [4], [5].

One approach to exploiting the correlation is to use source coding methods to compress the information before caching. Slepian-Wolf (SW) coding [6] leverages high degrees of correlation between information sources to compress their data in a distributed manner, without the need for collaborative communication. Although generally studied in the Wireless Sensor Network (WSN) setting, this technique has proven to be a promising solution to reduce the transmission rate in a caching scenario [7], [8].

A major limitation of this type of SW coding is that the decoding of the information is performed jointly, which

The associate editor coordinating the review of this manuscript and approving it for publication was Huaqing Li.

becomes computationally expensive when including multiple files. Thus, it is desirable to reduce the complexity of the coding without greatly sacrificing the compression gains. In general, this is referred to as the clustered SW problem [9]. One way to achieve this is to limit the number of files to compress, but selecting the most optimal subset of files in terms of compression gains becomes an NP-hard problem which is not addressed by Literature to date. As a result, this paper develops two well-performing but sub-optimal methods that find the best performing subset of files to compress.

A. BACKGROUND AND MOTIVATIONS

Maddah-Ali and Niesen [1] were the first to optimise caching in terms of the global caching gain, which is the total memory available at the user end. They introduced a novel coding technique, called Coded Caching (CC), that intelligently stores parts of all the files on the main server with the end users during the *placement* phase. When the users' requests are revealed in the *delivery* phase, the server is able to compress all the requests into a single multicast file based on the knowledge of the files stored previously. On receiving the multicast message, each end user reconstructs their requested file by XOR'ing the contents of their local cache. Much work has been done in this field, such as considering the fundamental coding limits in a case where the caches are shared between the users [10] and the optimal placement of files based on popularity [7], [8]. It has also found to be useful in other current research topics, like Information-Centric Networking (ICN), which replaces the traditional server/user model with information being stored in the network itself [11].

In the specific case where files are correlated with one another, Hassanzadeh et al. describe how the caching bounds could be improved [2]. This is achieved by dividing the files into different subsets. The subsets can be used to compress one another, since information will be repeated owing to the correlation. The compressed files are then stored at the caches and can be recovered after the users reveal their file requests. The compression has the effect that more of the information can be stored at the cache, minimising the size of the multicast message sent by the server during the delivery phase [3]. By simplifying the model, a more optimal placement scheme was designed in [4]. This was made more systematic by incorporating Gray-Wyner coding into the compression design [5]. However, as the authors note, the constraints on the bounds required to achieve optimal compression grow exponentially with the increase in the number of files. As such, they only present the cases where two files are transmitted to k users and three files are transmitted to two users.

Gray-Wyner coding is considered part of the field of Distributed Source Coding (DSC), where multiple independent pieces of information can be compressed at once, and the goal is to reduce bandwidth usage by limiting communication between users. SW coding falls under DSC and is similar to Gray-Wyner coding, with changes in the general structure [12]. In fact, Merikhi and Soleymani use the idea of decoding using side information (a feature of SW coding) in their CC implementation, although it is only used to compress two information sources [7], [8]. SW coding, too, suffers from exponential increases in coding constraints [13]. Furthermore, DSC schemes require jointly decoding the compressed information received, leading to increased complexity for the number of files included in the scheme. Wang et al. first proposed a method of dealing with this increase in complexity. They clustered the files into groups, such that the overall compression was maximised while the computational complexity was bounded [9]. They extended their work to increase the security [14] and overall compression [15] of their system. Shu, one of the authors of the above papers, used the same approach and added robustness by electing backup nodes [16] and energy efficiency by correlating the chance of becoming a cluster head to the distance from the next hop [17]. In a similar vein, Yang et al. [13], [18] provide a solution based on Lagrangian multiplier optimisations to organise the sources into a simpler structure, in an effort to reduce the SW decoding complexity. To this end, they simplify the correlation structure by only considering sources that are within a fixed radius of one another. This same simplification is used by Yuen et al. in [19], albeit with a different method to find the optimal results. More recently, Amutha et al. present this problem and use a sailfish meta-heuristic algorithm as a potential solution [20].

As a result, by adapting the clustering solution for SW coding to the CC domain, it is possible to simplify the system model in [5] and achieve the coding bounds for a greater number of files and users without increasing the complexity significantly. However, there are limitations to the current research around clustered SW as well. Firstly, in all the papers cited above, the solution involves calculating the performance of every combination of sources not yet selected. With the increase in the number of sources, this approach becomes computationally infeasible. Yang et al. reduce this complexity somewhat by disregarding correlations below a certain threshold. They also simplify the entropy calculation by modelling the correlation as a Gaussian distribution. Nevertheless, there are two issues with this approach. Firstly it is possible that, even with this simplification, there will still be many sources in the sensing radius if the source distribution is dense. Thus the original problem of complexity will return, since they do not fundamentally change the method for searching for the most optimal grouping. Secondly, using a fixed sensing radius and correlation model might grossly oversimplify the problem, since it is based on a simple spatial distance metric. This does not take into account correlation-specific metrics and will thus not be helpful in the CC domain, where the information sources are files to be compressed as opposed to the WSN consideration, where the information sources are nodes in a network.

B. CONTRIBUTIONS

Motivated by the gaps and shortcomings identified above, the main contributions of this paper are to:

1) Adapt the current work regarding SW coding in a WSN environment to the CC with correlated sources scenario. This involves using the optimisations of clustering for SW to simplify the system model given in CC for a Gray-Wyner network and thus reduce the complexity of the coding and decoding.
2) Incorporate a different model of the correlation between information sources into evaluating the entropy performance of the system. We choose to use the summation of Mutual Information Areas (MIAs) instead of Gaussian random variables, as these areas are independent and are able to be tailored to a variety of cases, including files in a CC setting.
3) Create two novel solutions to the clustered SW problem with CC considerations. This is done without relying on the simplifications provided in Literature to date.

The comparison of our work to other Literature across the different fields is presented in Table 1.

TABLE 1. Comparison of current works and our paper.

Reference	DSC	CC	Clustering	MIA Modelling
[5]	✓	✓	✗	✗
[7]	✗	✓	✗	✗
[9]	✓	✗	✓	✗
[13]	✓	✗	✓	✗
Our paper	✓	✓	✓	✓

C. LAYOUT

The rest of this paper is structured as follows: Section II describes the system model and optimisation problems, while Sections III and IV outline the two approaches. Section V analyses the complexity and optimality of the various solutions and Section VI presents and compares the simulation results. Section VII presents a brief discussion and outlines future work directions. Finally, Section VIII concludes this paper.

II. SYSTEM MODEL AND PROBLEM FORMULATION

The system consists of two primary components: CC and SW coding. The former ensures that the bandwidth of the server is minimised during peak hours while the latter seeks to reduce the total amount of information needed to be sent by the server to the users by compressing the files beforehand. A general outline of the system model is shown in Fig. 1. The following two subsections provide more detailed modelling for each subsystem.

A. CODED CACHING MODEL

In [7], [8], Merikhi and Soleymani present a CC system model in which users can receive information from the server or from shared remote caches. In contrast, this paper focuses on the single server case with local caches, where Z users connect to a single base station over an error-free broadcast link. The base station contains a library of files represented by the set $\mathbf{N}$. Each element of $\mathbf{N}$ is modelled as an information source $X_i, i \in \{1, 2, \ldots, |\mathbf{N}|\}$. Without loss of generality, each file X_i produces F binary symbols which are i.i.d and ergodic. Accordingly, each file has an entropy $H(X_i) = F$ bits, $\forall X_i \in \mathbf{N}$. However, it is assumed that the files are correlated to one another according to the distribution $p(x_1, x_2, \ldots, x_{|\mathbf{N}|})$. In addition, each user has at its disposal a local cache of size M files, or MF bits. We denote $\mathbf{Z}_i$ as the contents of user i's cache.

The CC system operates in two distinct phases. In the *placement* phase, the base station intelligently fills the users' caches with files from its library during off-peak hours. Thus, the transmission rate is not constrained in this phase, only the size of memory available at the user end.

In the *delivery* phase, the users reveal their demands, modelled here as a vector $\mathbf{d} := \{d_1, d_2, \ldots, d_Z\}$, where each d_i is an index corresponding to a request from user i for file X_{d_i}. The base station attempts to fulfil the file requests of the users by broadcasting a compressed version of the files, based on the placement in the caches in the previous phase.

As in [5], the objective of the caching scheme is evaluated according to the minimum multicast rate necessary to fulfil the worst demand:

$$R_m = \max_{\mathbf{d}\in\mathbf{D}} \frac{\mathbb{E}[\ell(Y_\mathbf{d})]}{F}, \tag{1}$$

where $\ell(\cdot)$ is the length of the broadcast codeword Y for demand $\mathbf{d}$, and $\mathbf{D}$ is the set of all possible demand vectors. Another evaluation metric is the average multicast rate, defined as:

$$\bar{R}_m = \frac{\mathbb{E}[\ell(Y_\mathbf{D})]}{F} \tag{2}$$

It is based on the average broadcast codeword length over all demands.

B. SW CODING MODEL

The correlation between the sources is modelled as the MIAs for each unique subset of $\mathbf{N}$. Thus, there are a total of $2^{|\mathbf{N}|} - 1$ areas. Naturally, the entropy of all the files $H(\mathbf{N}) \leq |\mathbf{N}|F$ bits. Nevertheless, the correlation values are merely statistical and do not necessarily describe the actual correlation between the contents of the files.

Since the files are correlated, the base station is able to use SW coding to compress the files that are stored in the caches, although the exact contents of the files are unknown. This has the effect of storing more content from the files in the users' caches, meaning that $\ell(Y_\mathbf{d})$ is reduced. However, there are two primary restrictions on this method.

Firstly, in terms of the compression itself, there are bounds given by Cover [21] for $|\mathbf{N}|$ sources. In total, there are $2^{|\mathbf{N}|} - 1$ bounds corresponding to each combination of sources. For

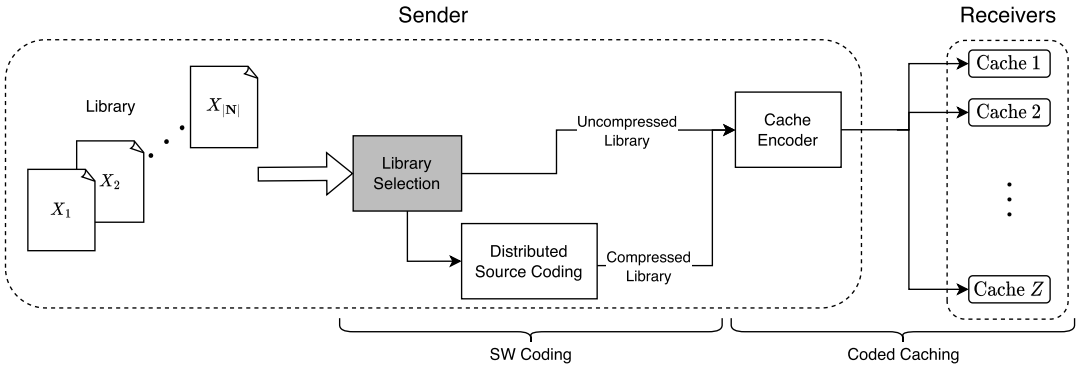

FIGURE 1. System model overview.

example, for 3 sources $\{X_1, X_2, X_3\}$, the 7 coding bounds are:

$$\begin{aligned} R_1 &\geq H(X_1|X_2, X_3) \\ R_2 &\geq H(X_2|X_1, X_3) \\ R_3 &\geq H(X_3|X_1, X_1) \\ R_1 + R_2 &\geq H(X_1, X_2|X_3) \\ R_1 + R_3 &\geq H(X_1, X_3|X_2) \\ R_2 + R_3 &\geq H(X_2, X_3|X_1) \\ R_1 + R_2 + R_3 &\geq H(X_1, X_2, X_3) \end{aligned}$$

Significantly, the bound on the total coding rate, and therefore the maximum compression of the system as a whole, is given as the joint entropy, which in general can be expanded to:

$$\sum_{i=1}^{|\mathbf{N}|} R_i \geq H(\mathbf{N}) = H(X_1, X_2, \ldots, X_{|\mathbf{N}|}) \tag{3}$$

$$= H(X_1) + \sum_{i=2}^{|\mathbf{N}|} H(X_i|X_{i-1}, X_{i-2}, \ldots, X_1) \tag{4}$$

Equation (4) is obtained by repeatedly using the chain rule for entropy.

Secondly, this bound is only achievable if all sources are decoded jointly (although the coding is disjoint). This joint decode is computationally expensive and is governed by the number of sources involved in the coding scheme.

Thus, the general optimisation problem is to decrease the complexity by removing sources from the scheme while minimising the impact on the achievable compression for the other files. As a result, this paper considers a system shown in Fig. 1, which is adapted from [5]. In that paper, all the files are compressed before they are transmitted to the caches. The goal of our system is to partition the library into two groups, one of which will be compressed according to a DSC method such as SW with Matrix Partitioning [22], denoted as $\mathbf{N}_c$. The other group will remain uncoded and is represented by $\mathbf{N}_u$. The sets are disjoint, meaning that $\mathbf{N}_c \cup \mathbf{N}_u = \mathbf{N}$ and $\mathbf{N}_c \cap \mathbf{N}_u = \emptyset$. This new hybrid library can then be distributed amongst the Z users using a cache encoder optimised for files of unequal lengths (such as [23]). Unlike [4], which considers that files are divided into a finite number of blocks, this paper allows for compressed files of any size.

We now turn to more formally defining the objectives of the system.

C. OBJECTIVES

The main goal of the hybrid system is to reduce the complexity of the encoding and decoding of the compression scheme without sacrificing too much of the compression rate for the system as a whole. However, there are two approaches to achieve this. In the first, the complexity is fixed by setting the maximum number of nodes to compress. Hence, let $\gamma = |\mathbf{N}_u|$ be the minimum number of nodes that should not be compressed, at which the number of compressed nodes $|\mathbf{N}_c| = |\mathbf{N}| - \gamma$ achieves a reasonable decoding complexity. Then, the objective is to choose a subset $\mathbf{N}_u$ from $\mathbf{N}$ that maximises the reduction in information for the coded sources $\mathbf{N} \setminus \mathbf{N}_u = \mathbf{N}_c$. This is formulated as follows:

$$\mathbf{N}_u^* = \arg\max_{\mathbf{N}_u} H(\mathbf{N}) - H(\mathbf{N}_u) \tag{5}$$

$$\text{s.t.} \quad \mathbf{N}_u \subset \mathbf{N}, |\mathbf{N}_u| = \gamma \tag{6}$$

Another approach is to bound the entropy of the compressed group of sources, effectively setting the compression performance in this group. Then, the objective is to find the maximum number of sources to put in $\mathbf{N}_c$ without exceeding the entropy bound. As a result, let ζ be an entropy value in the range $0 < \zeta < H(\mathbf{N})$. The goal is to minimise the number of sources in $\mathbf{N}_u$ while keeping the entropy of the compressed group to $H(\mathbf{N}_c) \leq \zeta$. In this instance, the objective function is defined as:

$$\mathbf{N}_u^* = \arg\min_{\mathbf{N}_u} |\mathbf{N}_u| \tag{7}$$

$$\text{s.t.} \quad H(\mathbf{N}_c) \leq \zeta \tag{8}$$

These optimisation problems are similar to the Minimum Weight Set Covering (MWSC) problem, where each subset

has a weight attached to it and the goal is to choose the fewest subsets that covers all members of the set while minimising the total weight. This problem is known to be NP-hard [24].

Another similar optimisation problem is the 0-1 Knapsack (0-1K) problem, in which a set of members each contain a weight and value. The objective is to choose the best performing subset of members that maximises the total value of the members while not exceeding a certain total weight. This problem too is NP-hard [25].

The optimisation problems in this paper are similar to the MWSC and 0-1K problems, but in those problems the number of combinations in the search space for a set of size $|\mathbf{N}|$ is $2^{|\mathbf{N}|}$, where in our scenario it is $\binom{|\mathbf{N}|}{\gamma}$. Nevertheless, in our case, the weights (entropies) of each subset are not known beforehand and must be calculated, unlike the MWSC and 0-1K problems. Furthermore, in our scenario, the weight calculation must be summed over $2^{\gamma} - 1$ areas. As a result, when γ is in the region $|\mathbf{N}|/2$, or if γ is large, our optimisation problems become NP-hard.

Accordingly, it is difficult to find an optimal solution in polynomial time. In the next sections, two sub-optimal approaches are discussed.

III. GREEDY ITERATIVE SELECTION PROCEDURE

The basic approach to solving the optimisation problem in (5) is to iteratively select the most suitable sources until a subset of size γ is reached. Further examination of the expansion of the total entropy in (4) reveals that the entropy of the set can be expressed in individual terms, where each term refers to only one source conditioned on other sources. This means that choosing the source X_i for each term, such that $H(X_i|X_{i-1}, X_{i-2}, \ldots, X_1) \geq H(X_j|X_{i-1}, X_{i-2}, \ldots, X_1)$, $\forall j \in \{i+1, \ldots, |\mathbf{N}|\}$ should guarantee a maximisation of the entropy expression at that point. The only exception is the first term, since, as mentioned in Section II-A, the entropies of all sources are set to be the same. In this case, it is necessary to choose the source based on a different criterion. Notice that the final term in (4) is $H(X_{|\mathbf{N}|}|X_{|\mathbf{N}|-1}, X_{|\mathbf{N}|-2}, \ldots, X_1)$, which is equivalent to the MIA $I(X_{|\mathbf{N}|}|X_{|\mathbf{N}|-1}; X_{|\mathbf{N}|-2}; \cdots; X_1)$. Thus, choosing the source that maximises the entropy term is equivalent to selecting the node that is least correlated with the other sources.

These observations imply that, by continually selecting the largest term from the available set of files to add to $\mathbf{N}_u$, the entropy of the set $\mathbf{N}_c = \mathbf{N} \setminus \mathbf{N}_u$ is minimised. As a result, the following selection procedure is proposed:

Lemma 1 (source selection procedure). Let the sources in $\mathbf{N}$ be drawn according to the following conventions: Choose X_1 such that $H(X_1|\mathbf{N} \setminus X_1) \geq H(X_i|\mathbf{N} \setminus X_i)\ \forall X_i \in \mathbf{N} \setminus X_1$. Then, for $k \in \{2, \ldots, |\mathbf{N}|\}$, choose source X_k such that

$$H(X_k|X_{k-1}, X_{k-2}, \ldots, X_1) \geq H(X_i|X_{k-1}, X_{k-2}, \ldots, X_1) \tag{9}$$

$$\forall X_i \in \mathbf{N} \setminus \{X_1, X_2, \ldots, X_k\} \tag{10}$$

It is possible for there to be multiple options for X_k, in which case X_k should be chosen arbitrarily.

Lemma 2. If the sources are organised as outlined in Lemma 1, then it is guaranteed that

$$H(X_k|X_{k-1}, X_{k-2}, \ldots, X_1) \geq H(X_{k+1}|X_k, X_{k-1}, \ldots, X_1) \tag{11}$$

Proof: From (9) it is known that

$$H(X_k|X_{k-1}, X_{k-2}, \ldots, X_1) \geq H(X_{k+1}|X_{k-1}, X_{k-2}, \ldots, X_1) \tag{12}$$

$$\geq H(X_{k+1}|X_k, X_{k-1}, \ldots, X_1) \tag{13}$$

Since conditioning reduces entropy. □

Using this selection procedure thus ensures that the terms in the expansion are in descending order.

Theorem 1. Let the sources be indexed according to Lemma 1. If, at any point in the selection process, $H(X_k|X_{k-1}, X_{k-2}, \ldots, X_1) \geq \zeta \geq H(X_{k+1}|X_k, X_{k-1}, \ldots, X_1)$ for some threshold ζ, then

$$H(\mathbf{N}) - H(X_1, X_2, \ldots, X_k) \leq (|\mathbf{N}| - k)\zeta \tag{14}$$

Proof: Since $\zeta \geq H(X_{k+1}|X_k, X_{k-1}, \ldots, X_1)$ then from Lemma 2

$$\zeta \geq H(X_{k+1}|X_k, X_{k-1}, \ldots, X_1) \tag{15}$$

$$\geq H(X_{k+2}|X_{k+1}, X_k, \ldots, X_1) \tag{16}$$

$$\vdots \tag{17}$$

$$\geq H(X_{|\mathbf{N}|}|X_{|\mathbf{N}|-1}, X_{|\mathbf{N}|-2}, \ldots, X_1) \tag{18}$$

Together with this, we have

$$H(\mathbf{N}) - H(X_1, X_2, \ldots, X_k) = \sum_{i=1}^{|\mathbf{N}|-k} H(X_{k+i}|X_{k+i-1}, X_{k+i-2}, \ldots, X_1) \tag{19}$$

Finally,

$$(|\mathbf{N}| - k)\zeta \geq \sum_{i=1}^{|\mathbf{N}|-k} H(X_{k+i}|X_{k+i-1}, X_{k+i-2}, \ldots, X_1) \tag{20}$$

□

Theorem 1 can thus be used as a stopping condition to satisfy (7). If, while choosing the sources, the entropy of the selected source is less than ζ, then the rest of the sources' entropy contribution can be bounded and the selection process will terminate with k sources.

The above derivations are combined to produce the Greedy Iterative Single Group Entropy Minimisation (GISGEM) selection procedure, outlined in Algorithm 1. It is able to generate a potential solution to the optimisation problems in (5) and (7) by maximising the entropy of the source removed at each step.

Algorithm 1 GISGEM Selection Procedure

$\mathbf{N}_c \leftarrow \mathbf{N}$
$\mathbf{N}_u \leftarrow \emptyset$
$i \leftarrow 1$
sort $\mathbf{N}_c$ s.t. $H(X_i|\mathbf{N} \setminus X_i) \geq H(X_{i+1}|\mathbf{N} \setminus X_{i+1}) \geq \cdots \geq H(X_{|\mathbf{N}_c|}|\mathbf{N} \setminus X_{|\mathbf{N}_c|})$
$\mathbf{N}_u \leftarrow \mathbf{N}_u \cup X_i$
$\mathbf{N}_c \leftarrow \mathbf{N}_c \setminus X_i$
while $|\mathbf{N}_u| \leq \gamma$ **do**
  $i \leftarrow i + 1$
  sort $\mathbf{N}_c$ s.t. $H(X_i|\mathbf{N}_u) \geq H(X_{i+1}|\mathbf{N}_u) \geq \cdots \geq H(X_{|\mathbf{N}_c|}|\mathbf{N}_u)$
  if $H(X_i|\mathbf{N}_u) < \zeta$ **then**
    Break
  end if
  $\mathbf{N}_u \leftarrow \mathbf{N}_u \cup X_i$
  $\mathbf{N}_c \leftarrow \mathbf{N}_c \setminus X_i$
end while
return $\mathbf{N}_c, \mathbf{N}_u$

IV. META-HEURISTIC APPROACH

The Genetic Algorithm (GA) is a meta-heuristic algorithm whose efficacy in combinatorial problems is well known [26]. Every valid combination is represented by a genome sequence, which sets the variables in the optimisation problem. For the single group entropy minimisation problem, the genome structure is updated as follows: the position in the genome represents the source in the set $\mathbf{N}$ and can take on the Boolean values 0 and 1 depending on whether the source is in the set $\mathbf{N}_u$ or not. This means that the genomes are of length $|\mathbf{N}|$ with the constraint that the Hamming weight must be equal to γ.

A population of random genomes is created, with rules defined for populating the next generation based on genome crossover and mutation. The crossover function selects features from both parents, based on a random crossover point. In our approach, the crossover functions need to produce child genomes that conform to the weight constraint mentioned above. As a result, the crossover function is changed to the following: given two genome sets $\mathbf{G_1}$ and $\mathbf{G_2}$ (defined as the sources in $\mathbf{N}_u$ shown by genome sequences $\mathbf{g_1}$ and $\mathbf{g_2}$), the child genome set $\mathbf{G_3}$ is constructed by randomly choosing γ sources from $\mathbf{G_1} \cup \mathbf{G_2}$.

The mutation function randomly flips a bit in the genome, ensuring that the genome pool does not become too small. Here, the mutation is updated to swap a random number of value pairs in the genome, since simply flipping a single bit could violate the Hamming weight constraint. The rest of the algorithm, such as parent selection, does not require any changes. A summary of this approach can be found in Algorithm 2.

V. ANALYSIS

The methods outlined above are now analysed in terms of the algorithm complexity and whether they produce objectively optimal solutions as compared to a Brute Force (BF) search.

Algorithm 2 Genetic Algorithm Approach

$\mathbf{P} \leftarrow$ random population of genomes
Calculate and rank the performance of each genome in $\mathbf{P}$
while not converged **do**
  Choose $\alpha_c|\mathbf{P}|$ random pairs of parent genomes for crossover
  for each parent genome pair $\{\mathbf{g_1}, \mathbf{g_2}\}$ **do**
    $\mathbf{G}_{\text{child}} \leftarrow \gamma$ random sources from $\mathbf{G_1} \cup \mathbf{G_2}$ ▷ Crossover
  end for
  Choose $\alpha_m|\mathbf{P}|$ random genomes for mutation
  for each genome $\mathbf{g_i}$ **do**
    $\mathbf{g}_{\text{child}} \leftarrow \mathbf{g_i}$
    swap random number of pairs in $\mathbf{g}_{\text{child}}$ ▷ Mutation
  end for
  $\mathbf{P}_{\text{new}} \leftarrow |\mathbf{P}|(1 - \alpha_c - \alpha_m)$ best genomes from $\mathbf{P}$
  $\mathbf{P}_{\text{new}} \cup$ crossover children
  $\mathbf{P}_{\text{new}} \cup$ mutation children
  $\mathbf{P} \leftarrow \mathbf{P}_{\text{new}}$
  Calculate and rank each genome in $\mathbf{P}$
end while

A. COMPLEXITY

The BF approach to solving the entropy minimisation optimisation requires that every combination of $\mathbf{N}_u$ is determined and evaluated and the grouping that provides the least impact to the compression gains is chosen. This means that the complexity is in the order of $O\left(\binom{|\mathbf{N}|}{\gamma}\right)$. In contrast to this, the GISGEM selection procedure has a much lower complexity. In the ith step, there are $|\mathbf{N}|-i-1$ terms that are calculated. Thus, the total number of calculations required for γ steps is

$$\gamma|\mathbf{N}| - \sum_{i=1}^{\gamma-1} i = \gamma(|\mathbf{N}| - 1/2(\gamma - 1)) \tag{21}$$

There is the additional complexity of ranking the terms in each step, however this complexity is negligible when compared to calculating the entropy. As a result, the complexity of this selection procedure is $O(\gamma|\mathbf{N}|)$. The complexity of the GA varies, and is dependent on the size of the population, as well as the number of generations required until the algorithm converges.

In Literature, the authors in [9] propose a greedy algorithm that requires ranking the power set of $\mathbf{N}$. However, their algorithm finds the minimum entropy disjoint grouping for *all* sources. Nevertheless, when determining the best grouping of size γ, this method will have a complexity in the same order as the BF method. Thus, the GISGEM selection procedure dramatically reduces the complexity as compared to the brute-force and Literature methods when γ is in the region of $|\mathbf{N}|/2$.

B. OPTIMALITY

Although less complex than the BF approach, the following Lemmas show that using GISGEM does not provide an

optimal result with regards to the optimisations in (5) and (7) respectively.

Lemma 3. The GISGEM selection procedure is not optimal in terms of the optimisation in (5).

Proof: Since the GISGEM selection procedure is iterative, it begins with $\mathbf{N}_u = \emptyset$ and adds a single source at a time. Without any stopping conditions, this will result in $\mathbf{N}_u = \mathbf{N}$. However, every possible order of choosing sources to put into $\mathbf{N}_u$ must follow these same start and end states. Thus, it is impossible that a single selection order will produce an optimal result for any arbitrary $0 \leq \gamma \leq |\mathbf{N}|$, since it cannot guarantee to be optimal at every point. □

Lemma 4. GISGEM is not optimal in terms of the optimisation in (7).

Proof: Theorem 1 provides a stopping condition to the GISGEM algorithm. Nevertheless, as shown in Lemma 3, this result is not necessarily optimal. Thus, it is possible that another selection of sources has a more optimal $\mathbf{N}_u$ and, even after removing one or more of the worst-performing sources, would still perform better than GISGEM. □

Thus, theoretically, the GISGEM algorithm should produce sub-optimal results but with less time complexity than the BF and GA methods. The following section details the numerical results obtained for the different approaches.

VI. RESULTS

In the following subsection, an illustrative example is presented, comparing the performance of the different coding schemes, namely the Coded Caching (CC) [1] and CC with SW (SW/CC) [5] approaches from Literature, and the hybrid method with Reduced Complexity (SW/CC-RC) proposed in this paper (depicted in Fig. 1). The methods used to reduce the complexity in the SW/CC-RC system are compared in the next subsection.

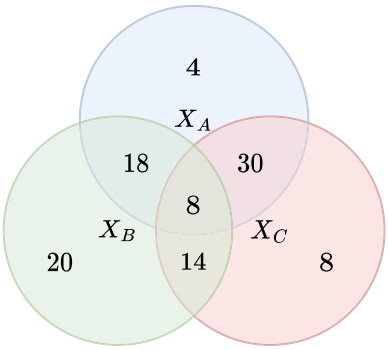

FIGURE 2. Example of a correlation diagram for three sources.

A. NUMERICAL EXAMPLE

Consider three files with a correlation model shown in Fig. 2, with each file having an entropy of 60 bits. Furthermore, let the number of users $Z = 2$, each with cache size $MF = 90$ bits. For the sake of clarity, we denote $X_A^{i:j}$ as representing a sub-file of X_A, consisting of bits i to j inclusive, where $i < j$ and files begin with bit 1. As a result, the length of this file $\ell(X_A^{i:j}) = j - i + 1$ bits. Furthermore, let $\{X_A^{i:j}; X_B^{k:l}\}$ (with a semi-colon) be the concatenation of sub-files from X_A and X_B.

In the classic CC method, where correlation is ignored, the ideal placement of files is to divide each file in half and place each half at a different user. When the users' demands are revealed, the server XOR's the files not currently stored by that user together and transmits the joint codeword. For instance, let the demand $\mathbf{d} = \{A, B\}$. Since user Z_1 already stored $X_A^{1:30}$ in its cache $\mathbf{Z}_1$ and user Z_2 has $X_B^{31:60}$ in $\mathbf{Z}_2$, the codeword transmitted by the server is $Y_{\mathbf{d}} = X_A^{31:60} \oplus X_B^{1:30}$. At the receiving end, the users XOR $Y_{\mathbf{d}}$ with the cached half of the file not requested to decode the rest of its requested file (e.g. Z_1 calculates $Y_{\mathbf{d}} \oplus X_B^{1:30} = X_A^{31:60}$). Under these conditions, it is readily verifiable that the peak demand rate $R_m = \bar{R}_m = 30$ bits.

The second approach, SW/CC, uses SW coding before doing the CC. To begin, it is necessary to calculate the bounds on the coding rate when compressing the files. In this example, there are 7 conditions to be met:

$$R_A \geq 4 \tag{22}$$
$$R_B \geq 20 \tag{23}$$
$$R_C \geq 8 \tag{24}$$
$$R_A + R_B \geq 42 \tag{25}$$
$$R_A + R_C \geq 42 \tag{26}$$
$$R_B + R_C \geq 42 \tag{27}$$
$$R_A + R_B + R_C \geq 102 \tag{28}$$

The most restrictive of these is (28). Thus, to satisfy all conditions, it is sufficient to set all coding rates to $102/3 = 34$ bits. Let the files compressed using these compression rates be $\hat{X}_A$, $\hat{X}_B$ and $\hat{X}_C$ (this can be achieved using the Matrix Partitioning method for SW coding [22]). Using the same cache size as above, we are now able to store 30 bits from each of these files, corresponding to 88% of the file as opposed to 50% in the CC method. In this case, both caches receive $\hat{X}_A^{1:30}$, $\hat{X}_B^{1:30}$, $\hat{X}_C^{1:30}$. The remaining 12 bits (4 from each file) are transmitted regardless of the demand vector.[1] On the receiver's end, the new information is used to perform a joint decode to losslessly obtain X_A, X_B and X_C. As a result, the performance is $R_m = \bar{R}_m = 12$ bits. However, this comes at an increased complexity cost, owing to the number of constraints that need to be met, as well as the joint decode complexity.

Finally, the SW/CC-RC method is demonstrated. Before performing the SW coding, the GISGEM algorithm is applied to the files to determine which ones to remove. Table 2 lists all the iterations used to index the files. In the first step, the excess entropies for each file (i.e. the entropy of each file conditioned on all other files) is calculated and the maximum

[1] In this specific example, there is no need for the CC method, since the cache sizes are large enough to contain sufficient information to negate the extra coding.

is chosen to be X_1. The next iterations exclude all previously chosen sources and the conditional entropies are calculated, with the maximum chosen in each successive stage. In this example, the selection procedure is always optimal, since the first and last steps are guaranteed to be optimal and there are only 3 iterations.

TABLE 2. Indexing of sources using GISGEM.

Source	$H(X_i\|\mathbf{N}\setminus X_i)$	$H(X_i\|X_1)$	$H(X_i\|X_2,X_1)$
X_B	20		
X_C	8	38	
X_A	4	34	4

If we set $\gamma = 1$ for example, the algorithm will suggest that X_B is removed. Thus, the library is partitioned into two subsets, $\mathbf{N}_c = \{X_A, X_C\}$ and $\mathbf{N}_u = \{X_B\}$. Consequently, the SW coding method from the previous SW/CC example will only be applied to the two files in $\mathbf{N}_c$. These files have 3 conditions for lossless SW coding:

$$R_A \geq 4 \tag{29}$$
$$R_C \geq 8 \tag{30}$$
$$R_A + R_C \geq H(X_A, X_C|X_B) = H(X_A, X_C) = 82 \tag{31}$$

Notice that the bound (31) is different to (26), since X_B is treated as independent of X_A and X_C in this case. Setting $R_A = R_C = 41$ bits satisfies all requirements and the files are compressed accordingly. Thus, the library now consists of $\{\hat{X}_A, X_B, \hat{X}_C\}$. In the next phase, the cache encoder stores 28 bits from the two compressed files and 34 bits from the uncompressed file. The caches are filled this way to balance out the peak multicast rate. The results of this phase are shown in Fig. 3, with $\mathbf{Z}_1 = \{\hat{X}_A^{1:28}, X_B^{1:34}, \hat{X}_C^{1:28}\}$ and $\mathbf{Z}_2 = \{\hat{X}_A^{14:41}, X_B^{27:60}, \hat{X}_C^{14:41}\}$. Using this placement, all files can be decoded at the user end regardless of $\mathbf{d}$, as shown in Table 3. In terms of performance, Table 3 also shows that $R_m = \bar{R}_m = 26$.

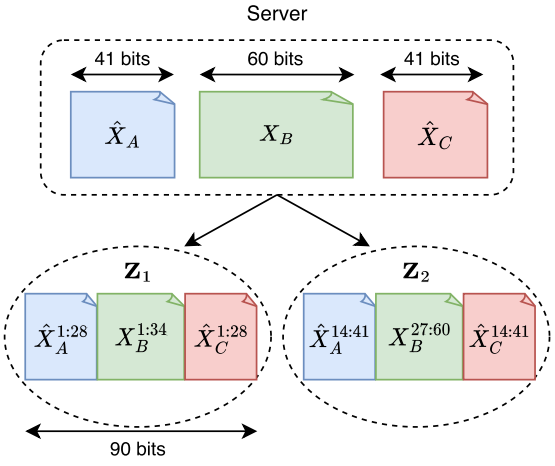

FIGURE 3. Cache storage contents for $|\mathbf{N}| = 3$ files, $Z = 2$ users and $MF = 90$ bits using the SW/CC-RC method.

In summary, Table 4 shows what is stored at each cache for the different approaches. Although the SW-CC method has the lowest average and peak rates, it comes at the cost of higher complexity in designing the coding scheme as well as the decoding algorithm. On the other hand, the SW/CC-RC method sacrifices some of the compression gains to achieve reduced complexity in terms of the number of equations, the decoding algorithm and the number of demand permutations in which SW decoding is needed at each cache (6 out of 12 permutations, compared to 12 for the SW/CC method). Nevertheless, the average rate is still reduced as compared to the regular CC approach.

B. SIMULATION RESULTS

Three methods have been outlined in this paper to determine the sources to remove in the SW/CC-RC approach. These are the BF, GA and GISGEM methods. Literature does not deal directly with the optimisation problem presented in this paper, and the current approaches (when ignoring their simplifications) are equivalent to the BF method.

To perform the comparison, 18 sources were used, where the optimal grouping was found for an increasing number of sources in $\mathbf{N}_u$. Fig. 4 shows the entropy obtained for each method when increasing the size of the selected group. Each of the methods are compared to the worst performing configuration (found using the BF method). The results confirm the sub-optimal performance predicted in Lemma 3, as it is found that, in the range of best to worst performing results, the GISGEM's results fall in the 91st percentile on average, with a minimum value falling in the 82nd percentile. The GA's results are in the 92nd percentile on average with the lowest value in the 76th percentile. As compared to the BF method, both the GISGEM and GA are able to find results close to the optimal one. However, since the GA is not constrained to choosing the same grouping as the previous iteration, it is sometimes able to find a better result than GISGEM. Fig. 4 also shows the correctness of Theorem 1, since the distance between the most and least optimal group selections decreases as more sources are selected. This highlights another advantage of GISGEM, as it is able to terminate earlier if it detects that the current number of sources is sufficient to achieve the objective in (5).

The time taken to find the optimal result for each method is given in Fig. 5. The GA is, on average, 4.42 times faster than the BF exhaustive search, while GISGEM is 10.20 times faster and increases linearly with respect to γ. These practical results conform well to the theoretical complexity predictions in Section V-A.

In another simulation, the GISGEM and GA methods were tested, this time in a system with $|\mathbf{N}| = 22$. At this number of nodes, the BF method's complexity becomes prohibitively large. In Fig. 6, the difference between the total entropies of the uncoded groups $\mathbf{N}_u$ for the GISGEM and GA algorithms are plotted. It shows that, in this run, the GISGEM

TABLE 3. Encoding and decoding procedure for the SW/CC-RC system.

$\mathbf{d}$	Encoding ($Y_{\mathbf{d}}$)	$\ell(Y_{\mathbf{d}})$	Decoding for Z_1	Decoding for Z_2
$\{A,A\}$ $\{A,C\}$ $\{C,A\}$ $\{C,C\}$	$\{\hat{X}_A^{29:41};\hat{X}_C^{29:41}\}\oplus\{\hat{X}_A^{1:13};\hat{X}_C^{1:13}\}$	26	$Y_{\mathbf{d}}\oplus\{\hat{X}_A^{1:13};\hat{X}_C^{1:13}\}$ + SW decoding	$Y_{\mathbf{d}}\oplus\{\hat{X}_A^{29:41};\hat{X}_C^{29:41}\}$ + SW decoding
$\{A,B\}$ $\{C,B\}$	$\{\hat{X}_A^{29:41};\hat{X}_C^{29:41}\}\oplus X_B^{1:26}$	26	$Y_{\mathbf{d}}\oplus X_B^{1:26}$ + SW decoding	$Y_{\mathbf{d}}\oplus\{\hat{X}_A^{29:41};\hat{X}_C^{29:41}\}$
$\{B,A\}$ $\{B,C\}$	$X_B^{35:60}\oplus\{\hat{X}_A^{1:13};\hat{X}_C^{1:13}\}$	26	$Y_{\mathbf{d}}\oplus\{\hat{X}_A^{1:13};\hat{X}_C^{1:13}\}$	$Y_{\mathbf{d}}\oplus X_B^{35:60}$ + SW decoding
$\{B,B\}$	$X_B^{35:60}\oplus X_B^{1:26}$	26	$Y_{\mathbf{d}}\oplus X_B^{1:26}$	$Y_{\mathbf{d}}\oplus X_B^{35:60}$

TABLE 4. Cache contents for different coding methods.

	$\mathbf{Z}_1$	$\mathbf{Z}_2$
CC	$X_A^{1:30}, X_B^{1:30}, X_C^{1:30}$	$X_A^{31:60}, X_B^{31:60}, X_C^{31:60}$
SW/CC	$\hat{X}_A^{1:30}, \hat{X}_B^{1:30}, \hat{X}_C^{1:30}$	$\hat{X}_A^{1:30}, \hat{X}_B^{1:30}, \hat{X}_C^{1:30}$
SW/CC-RC	$\hat{X}_A^{1:28}, X_B^{1:34}, \hat{X}_C^{1:28}$	$\hat{X}_A^{14:41}, X_B^{27:60}, \hat{X}_C^{14:41}$

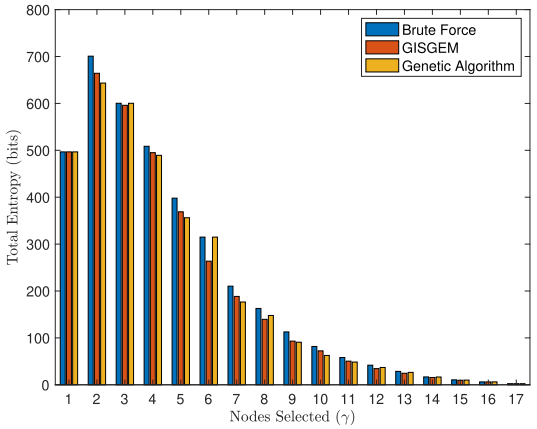

FIGURE 4. The performance of each selection method as compared to the least optimal combination for $|\mathbf{N}| = 18$.

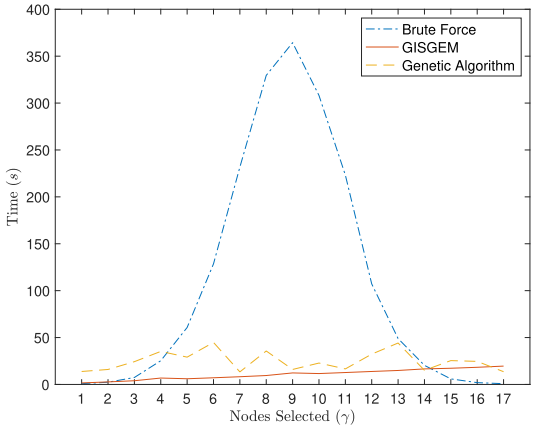

FIGURE 5. The time complexity of each selection method for $|\mathbf{N}| = 18$.

method outperforms the GA approach for small γ. For larger values of γ the two approaches are closer, with the GA approach sometimes besting the GISGEM one. Nevertheless, as depicted in Fig. 7, the time complexity of GISGEM is consistently lower than the moving average of the GA.

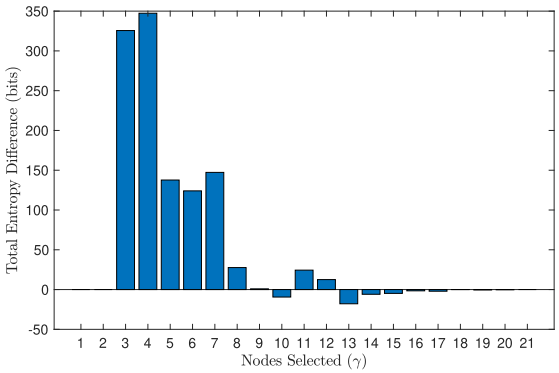

FIGURE 6. The difference between total entropies of $\mathbf{N}_u$ produced by the GISGEM and GA algorithms for $|\mathbf{N}| = 22$.

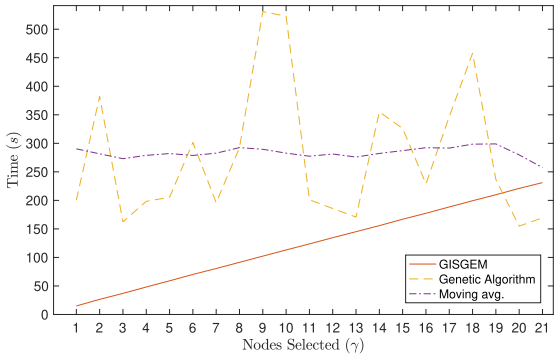

FIGURE 7. The time complexity of the GISGEM and GA selection methods for $|\mathbf{N}| = 22$.

C. GA PARAMETER TUNING

The GA has different parameters that can be tuned to obtain better results. As discussed in [27], these variables are critical to changing the diversification (breadth of search area) and intensification (refining the current results) of the algorithm. They also found that there is not necessarily a single parameter that controls each type of behaviour of the

TABLE 5. Different parameter settings for GA testing and their impact on the quality of results and complexity.

Parameter	Symbol	Range	Performance Impact	Time Impact	Iteration Impact
Stall Generations	SG	5, 10, 15, 20	Large	Large	Large
Population Size	$\|\mathbf{P}\|$	10, 15, 20	Medium	Small	Minimal
Elite Percentage	α_e	10%, 20%, 30%	Minimal	Small	Small
Crossover Percentage	α_c	20%, 40%, 60%, 80%	Small	Minimal	Minimal

algorithm. In order to examine the effect of each parameter in our scenario, the GA was run with parameters in the ranges shown in Table 5. To explain the parameters: SG refers to the number of iterations in which the the elite value obtained remains the same, after which the algorithm converges. The Population Size is the number of configurations tested in each generation. The value α_e is the number of members of $\mathbf{P}$ that survives to the next generation, expressed in terms of the percentage of $|\mathbf{P}|$. The Crossover Percentage is the number of children produced using the gene crossover method. It is expressed as the percentage of $|\mathbf{P}| - \alpha_e$. The number of mutation children is not shown here, as it is automatically set as the inverse of α_c.

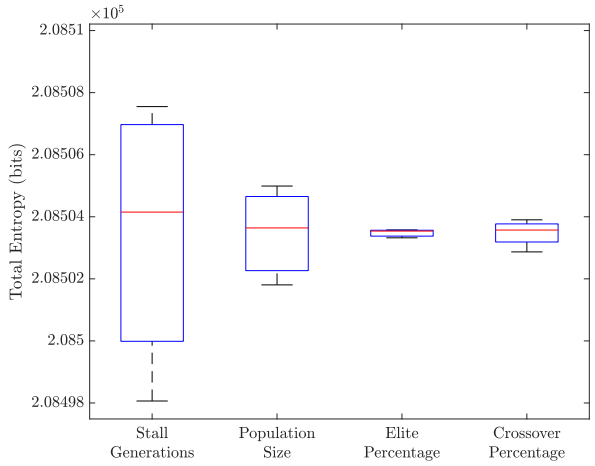

FIGURE 8. The effect of different GA configurations on performance.

The GA was run for $|\mathbf{N}| = 18$ and $\gamma \in \{7, 8, \ldots, 11\}$ for every combination of parameter. Figures 8-10 show the resulting effects, on average, of each parameter on the best result obtained, the time taken to converge as well as the number of iterations. These results are also summarised in Table 5. It is clear from the results that the SG has the biggest impact on both quality of results and time complexity. The trade-off between the two is found to be directly proportional. Consequently, an SG of 15 is chosen to slightly favour the quality of results over time complexity. Population size has a medium impact on quality but small impact on time complexity, while the number of iterations is barely affected. This is because increasing the population size increases the number of configurations tested per generation. For these reasons, the maximum population size of $|\mathbf{P}| = 20$ is chosen. The Elite Percentage has a minimal impact on quality of results but a small impact on time complexity. However, the results are found to not be linear, with $\alpha_e = 20\%$ producing the best time relative to the quality of results. Thus, this value is chosen for the final testing. Finally, the balance between α_c and α_m corresponds to a small impact in quality but minimal impact on time. It is found that $\alpha_c = 60\%$ maximises the best results obtained, so this is the value selected.

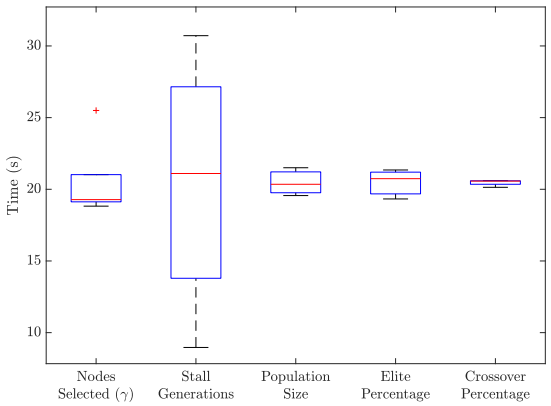

FIGURE 9. The effect of different GA configurations on convergence: time.

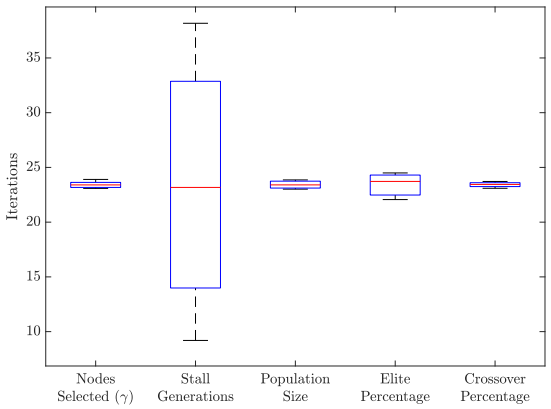

FIGURE 10. The effect of different GA configurations on convergence: iterations.

One interesting result is that a change in γ had a small effect on the time and minimal effect on the number of iterations taken to converge. This correlates with the head-to-head testing conducted in Section VI-B, where the moving average of the GA did not change much with a change in γ.

VII. DISCUSSION AND FUTURE WORK

Although the GA is suited to combinatorial problems in general, it is possible for a different meta-heuristic algorithm

(such as the Particle Swarm Optimisation or Binary Bat Algorithm) to perform better when considering the SW/CC-RC scenario. So too, although the parameters were tuned for this problem, tuning the functions of crossover and mutation might produce better results. Although this is a potential possibility for future work, it is noted that this paper uses the GA mainly as a benchmark with which to evaluate the performance of the GISGEM method and not as a stand-alone solution.

With regards to the GISGEM approach, the current design uses a 'no regret' scheme, where sources selected previously are not able to be removed in a future iteration. However, a more optimal result could be achieved by using a 'look ahead' approach and is a potential avenue for future work.

In addition, further work is necessary to generalise the method to solve the optimal clustering for all sources. In contrast to the single compressed group presented in this paper, the general grouping case allows for many groups of nodes, all of which will be compressed. In the single group scenario, the size of the search space is given by $\binom{|\mathbf{N}|}{\gamma}$. However, for the multiple grouping scenario, an expression that gives the size of the search space needs to be derived. Furthermore, the objective functions need to be changed to reflect the new considerations, where the entire library is able to be grouped, and the sum of the entropies of the groups needs to be minimised. It is also unknown how the approaches proposed in this paper will perform with the new considerations. Finally, it is surmised that the generalised scenario will result in an increase of compression with commensurate increase in complexity. However, it is necessary to compare this trade-off with the current results related to the system design in this paper.

VIII. CONCLUSION

A greedy iterative selection procedure and meta-heuristic approach are proposed as potential solutions to the clustered SW problem in the context of caching correlated information. Files are grouped together, such that the overall decoding complexity is reduced with minimal impact on compression gains. The iterative method is based on the inherent properties of entropy and it is able to find a close-to-optimal result with less time complexity as compared to the BF and meta-heuristic approaches. There is the additional benefit that the algorithm can bound the entropy gains at each iteration and can compare that to the relative complexity of the system, allowing it to terminate prematurely if necessary. It is found that these methods are able to successfully reduce the complexity of a SW/CC system while not sacrificing too much of the compression gains.

REFERENCES

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[2] P. Hassanzadeh, A. Tulino, J. Llorca, and E. Erkip, "Cache-aided coded multicast for correlated sources," in *Proc. 9th Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, Sep. 2016, pp. 360–364.

[3] P. Hassanzadeh, A. Tulino, J. Llorca, and E. Erkip, "Correlation-aware distributed caching and coded delivery," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Sep. 2016, pp. 166–170.

[4] K. Wan, D. Tuninetti, M. Ji, and G. Caire, "On coded caching with correlated files," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 692–696.

[5] P. Hassanzadeh, A. M. Tulino, J. Llorca, and E. Erkip, "Rate-memory trade-off for caching and delivery of correlated sources," *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 2219–2251, Apr. 2020.

[6] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 4, pp. 471–480, Jul. 1973.

[7] B. Merikhi and M. R. Soleymani, "Cache-aided delivery network in a shared cache framework with correlated sources," in *Proc. 18th ACM Int. Symp. QoS Secur. Wireless Mobile Netw.* New York, NY, USA: Association for Computing Machinery, 2022, pp. 121–129, doi: 10.1145/3551661.3561372.

[8] B. Merikhi and M. R. Soleymani, "Cache-aided networks with shared caches and correlated content under non-uniform demands," in *Proc. IEEE 20th Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2023, pp. 301–304.

[9] P. Wang, C. Li, and J. Zheng, "Distributed data aggregation using clustered Slepian–Wolf coding in wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2007, pp. 3616–3622.

[10] F. Brunero and P. Elia, "Fundamental limits of combinatorial multi-access caching," *IEEE Trans. Inf. Theory*, vol. 69, no. 2, pp. 1037–1056, Feb. 2023.

[11] K. Zheng, Y. Cui, X. Liu, X. Wang, X. Jiang, and J. Tian, "Asymptotic analysis of inhomogeneous information-centric wireless networks with infrastructure support," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5245–5259, Jun. 2018.

[12] R. Graczyk and A. Lapidoth, "Gray-Wyner and Slepian–Wolf guessing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 2189–2193.

[13] Y. Yang, S. Guo, G. Liu, and Q. Wang, "Joint source coding rate allocation and flow scheduling for data aggregation in collaborative sensing networks," *Comput. Netw.*, vol. 175, Jul. 2020, Art. no. 107269.

[14] P. Wang, C. Li, and J. Zheng, "Combined data aggregation and encryption using clustered Slepian–Wolf coding for wireless sensor networks," in *Proc. IEEE GLOBECOM Global Telecommun. Conf.*, Nov. 2007, pp. 920–925.

[15] J. Zheng, P. Wang, and C. Li, "Distributed data aggregation using Slepian–Wolf coding in cluster-based wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 5, pp. 2564–2574, Jun. 2010.

[16] Q. Shu, Q. Hu, J. Zheng, and N. Mitton, "A dependable Slepian–Wolf coding based clustering algorithm for data aggregation in wireless sensor networks," in *Proc. Int. Conf. Wireless Commun. Signal Process.*, Oct. 2013, pp. 1–6.

[17] Z. Huang and J. Zheng, "A Slepian–Wolf coding based energy-efficient clustering algorithm for data aggregation in wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 198–202.

[18] Y. Yang, S. Guo, and Y. Yang, "Distributed optimal source coding rate allocation for data aggregation in wireless sensor networks," in *Proc. IEEE 22nd Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2016, pp. 32–39.

[19] K. Yuen, B. Liang, and L. Baochun, "A distributed framework for correlated data gathering in sensor networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 1, pp. 578–593, Jan. 2008.

[20] R. Amutha, G. G. Sivasankari, and K. R. Venugopal, "Node clustering and data aggregation in wireless sensor network using sailfish optimization," *Multimedia Tools Appl.*, vol. 82, no. 28, pp. 44107–44122, Nov. 2023, doi: 10.1007/s11042-023-15225-z.

[21] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Hoboken, NJ, USA: Wiley, 2006.

[22] D. Schonberg, K. Ramchandran, and S. S. Pradhan, "Distributed code constructions for the entire Slepian–Wolf rate region for arbitrarily correlated sources," in *Proc. Data Compress. Conf.*, Mar. 2004, pp. 292–301.

[23] H. Cheng, C. Li, H. Xiong, and P. Frossard, "Optimal decentralized coded caching for heterogeneous files," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2017, pp. 2531–2535.

[24] Q. Xie, Y. Li, S. Hu, Y. Zhu, and H. Wang, "Two heuristic algorithms for the minimum weighted connected vertex cover problem under greedy strategy," *IEEE Access*, vol. 10, pp. 116467–116472, 2022.

[25] Z. Luo, X. Guo, B. Wen, and J. Cao, "An evolution algorithm based on cloud model for 0-1 knapsack problem," in *Proc. 15th Int. Symp. Comput. Intell. Design (ISCID)*, Dec. 2022, pp. 147–150.
[26] S. Forrest, "Genetic algorithms," *ACM Comput. Surv.*, vol. 28, no. 1, pp. 77–80, Mar. 1996, doi: 10.1145/234313.234350.
[27] A. Scheibenpflug and S. Wagner, "An analysis of the intensification and diversification behavior of different operators for genetic algorithms," in *Computer Aided Systems Theory EUROCAST*, R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, Eds. Berlin, Germany: Springer, 2013, pp. 364–371.

BENJAMIN ROSEN received the B.Eng.Sc. degree in digital arts and the B.Sc.Eng. degree in information engineering from the University of the Witwatersrand, Johannesburg, South Africa, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree.

His research interests include information theory, the IoT, wireless sensor networks, and source coding.

LING CHENG (Senior Member, IEEE) received the B.Eng. degree (cum laude) in electronics and information from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995, the M.Ing. degree (cum laude) in electrical and electronics, in 2005, and the D.Ing. degree in electrical and electronics from the University of Johannesburg (UJ), Johannesburg, South Africa, in 2011.

In 2010, he joined the University of the Witwatersrand, Johannesburg, where he was promoted to Full Professor, in 2019. He has been a Visiting Professor with five universities and the Principal Advisor for more than 50 full research postgraduate students (11 Ph.D. graduates). He has authored or coauthored more than 150 research papers in journals and conference proceedings. His research interests include telecommunications and artificial intelligence.

Dr. Cheng was a recipient of the Chancellors Medals, in 2005 and 2019; and the National Research Foundation ratings, in 2014 and 2020. The IEEE ISPLC 2015 Best Student Paper Award was made to his Ph.D. student in Austin. He is the Vice-Chair of the IEEE South African Information Theory Chapter. He serves as an associate editor for three journals.