****

**INF8770 – Technologies multimédias**

**Automne 2024**

**Travail pratique #1 – Comparaison et caractérisation de méthodes de codage**

**Groupe 02 – Équipe A37**

**2221053 – Thomas Rouleau**

**Soumis à : M. Yassine Achour**

**2024-10-02**

**Travail à réaliser** : Caractériser deux méthodes de codage et déterminer quelle méthode de codage est la meilleure et dans quelles circonstances. Les réponses aux questions doivent se retrouver dans un rapport. Les codes informatiques utilisés doivent implémenter les méthodes avec les mêmes algorithmes que ceux vus en classe.

**Méthodes de codage étudiées** : ***Codage LZ77*** et ***Codage Arithmétique***

## Question 1 (/4)

Formulez une liste d’hypothèses à tester pour caractériser la performance des deux méthodes. Expliquez vos différentes hypothèses. Exemple : La méthode 1 donnera un code plus compact que la méthode 2 si le message...

* Qualité évaluée : 3.1 Formuler des hypothèses testables
  + Critère d’évaluation : Qualité et exhaustivité des hypothèses sur la performance comparative des deux méthodes. Les hypothèses doivent être bien expliquées.

Hypothèse 1 : Taux de compression

* Définition : Le rapport entre la taille des données compressées et la taille des données originales. Un taux de compression plus faible indique une meilleure performance en termes de réduction de taille.
* Mesure : Taux de compression=Taille du fichier compresse/Taille du fichier original

Puisque le codage arithmétique est une méthode probabiliste basée sur la théorie de l’entropie, il est efficace pour représenter l’information de manière compacte, peu importe la taille de l’information initial (<https://www.researchgate.net/publication/342629882_Comparison_of_Entropy_and_Dictionary_Based_Text_Compression_in_English_German_French_Italian_Czech_Hungarian_Finnish_and_Croatian>). En comparaison, la méthode LZ77 présente d’excellentes performances lors du codage de données répétitives, mais n’atteint donc pas toujours des taux de compression aussi élevés que la précédentes (<https://www.researchgate.net/publication/342629882_Comparison_of_Entropy_and_Dictionary_Based_Text_Compression_in_English_German_French_Italian_Czech_Hungarian_Finnish_and_Croatian>). Ainsi, nous formulons l’hypothèse que le codage arithmétique produira en moyenne un taux de compression supérieur à celui de LZ77

Hypothèse 2 : Temps de compression

* Définition : Le temps requis pour compresser un fichier. Il inclut le temps de traitement algorithmique et l'accès aux données.
* Mesure : Temps écoulé pour terminer la compression d’un fichier (en secondes ou millisecondes).

Étant donné que la méthode LZ77 repose sur des opérations telles que la recherche dans une fenêtre glissante et l'encodage basé sur un dictionnaire, tandis que le codage arithmétique utilise des calculs complexes d'estimation probabiliste des symboles et d'encodage entropique, ce dernier est limité dans sa capacité à coder plusieurs symboles répétés simultanément (<https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=8409000&fileOId=8518513>). Par conséquent, nous supposons que la méthode LZ77 nécessitera moins de temps pour compresser des données, en moyenne, comparativement au codage arithmétique.

The compression speed is faster compared to statistical methods since several symbols can be encoded at a time.

In order to optimize the solution, there are three aspects, compression ratio, compression speed, and hardware cost, which need to be explored when implementing the LZ77 algorithm. There are six parameters affecting these three performances. The parameters are dictionary size, look-ahead buffer size, hash table size, minimum match length, minimum move length, and the length of code word. In this part, an analysis on the influence of these parameters on the performances is discussed.

 Dictionary Size

The larger the dictionary size is, the more the match symbols can be found. Thus the compression ratio can be improved with the increased dictionary size. However, the increased dictionary size will slow down the compression speed.

<https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=8409000&fileOId=8518513>

Hypothèse 3 : memory usage/complexité mémoire/usage mémoire/ hardware cost/ CPU usasge)

* Définition : La quantité de mémoire utilisée pendant le processus de compression (ou de décompression). Cela inclut la mémoire allouée pour stocker des tables, des fenêtres de recherche, ou des structures de données utilisées par l'algorithme.
* Mesure : Utilisation de la mémoire pendant le processus (en Mo, Go).

Le codage arithmétique repose sur de nombreux calculs probabilistes, ce qui implique une demande élevée en ressources de calcul (CPU). Ce type de codage utilise des techniques avancées pour représenter les symboles en fonction de leurs probabilités, entraînant une plus grande complexité algorithmique ([**https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7428525**](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7428525)**)**. En revanche, la méthode LZ77 se base sur une fenêtre glissante et un dictionnaire pour identifier et encoder les séquences répétées de manière plus simple. Cela peut engendrer une utilisation accrue de la mémoire, particulièrement pour gérer les données dans la fenêtre et le dictionnaire, en fonction de leur taille ([**https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7428525**](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7428525)**)**. Ainsi, nous supposons que la méthode de codage arithmétique exigera une plus grand capacité CPU que la méthode LZ77, tandis que LZ77 demandera une plus grande utilisation de la mémoire pour le processus de compression

[**https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7428525**](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7428525)

**A table with numbers and text

Description automatically generated**

<https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=8409000&fileOId=8518513>

<https://tarsa.github.io/lossless-benchmark/#results>

<https://www.mattmahoney.net/dc/text.html>

<https://en.wikipedia.org/wiki/Lossless_compression>

<https://medium.com/@thusharabandara/measure-the-compression-performance-of-an-image-compression-algorithm-ea68c1839ec6>

<https://www.chipestimate.com/An-Introduction-to-Measuring-Compression-Performance/CAST/Technical-Article/2022/05/03>

<https://www.researchgate.net/publication/342629882_Comparison_of_Entropy_and_Dictionary_Based_Text_Compression_in_English_German_French_Italian_Czech_Hungarian_Finnish_and_Croatian>

## Question 2 (/4)

Décrivez en détails les expériences que vous allez réaliser pour vérifier les hypothèses formulées (bases de données utilisées, contenu des messages, critère d’évaluation, code informatique utilisé, etc.). Les expériences doivent être menées sur deux des trois types de données suivantes : Fichier WAV, Fichier texte (Chaîne de caractères significativement longue) ou Images de formats quelconques (JPEG, PNG, ...)

* Qualité évaluée : 3.3 Planifier et préparer des essais
  + Critère d’évaluation : Qualité de la description et de la justification des expériences, du choix des données et des programmes informatiques qui seront utilisés pour tester les hypothèses.

Pour vérifier les hypothèses formulées concernant les caractéristiques de performances des méthodes de compression LZ77 et codage arithmétique, nous allons nous concentrer sur deux types de données : les fichiers textes et les images. Ces deux catégories offrent une variété de structures de données permettant de vérifier les performances des algorithmes.

Les fichiers texte choisis comporteront des chaînes de caractères significativement longues, comportant de nombreuses répétitions de mots ou de séquences de caractères. Cela permettra de tester les capacités de compression des deux méthodes sur des données hautement redondantes. Par exemple, des textes littéraires comme « Les Misérables » ou des documents techniques seront utilisés. La longueur des fichiers variera, incluant des petits fichiers de 10 000 caractères à des textes plus volumineux dépassant le million de caractères. Les images utilisées seront soit en niveaux de gris, soit en couleurs (RGB), et varieront en termes de résolution (de petites images à haute définition). L’expérimentation portera sur des images simples (comme des logos avec peu de détails) et des images complexes (comme des photos avec des dégradés). Le format PNG sera particulièrement intéressant pour tester l’effet de la compression sur des fichiers déjà compressés sans perte, tandis que les fichiers JPEG permettront de comparer les performances sur des données partiellement compressées. Ces choix permettent de tester les méthodes de compression sur des types de données avec des caractéristiques distinctes.

Pour le taux de compression, nous comparerons les méthodes LZ77 et codage arithmétique en mesurant la tailles de fichiers originaux et compressées. Les fichiers texte et les images seront compressés à l'aide des deux méthodes, et nous calculerons le taux de compression en utilisant la formule de gain ou d’économie d’espace :

Le temps de compression sera mesuré pour chaque fichier. Nous mesurerons le temps CPU nécessaire à la compression par la somme du temps utilisateurs et système rapporté par la commande Shell Unix time. Ainsi, seulement le temps CPU sera rapporté en ignorant les temps d’inactivités tel que l’attente d’opérations sur le disque.

L’utilisation de la mémoire et du CPU sera également mesuré pour chaque fichier compressé afin d’obtenir une moyenne représentative de chaque algorithme de compression. Nous utiliserons la **librairie python psutil** **ou top** afin d’obtenir une moyenne d’utilisation des ressources lors de la compression des données.

Les mêmes fichiers texte et images seront utilisés pour tous les tests afin de garantir leurs cohérences. Nous utiliserons des implémentations des algorithmes LZ77 et codage arithmétique en **Python** pour effectuer ces compressions. Les tests seront effectués sur Windows Subsytem for Linu9x (WSL) ayant une image ubuntu 22.04. **L’ordinateur utilisés pour les tests possède un processeur Intel i9-13900H de 14 cœurs et 32 GB de mémoire vive.**

Question 3 (/4)

Décrivez les codes informatiques utilisés pour réaliser les expériences. Décrivez, si applicable, comment vous avez adapté les codes informatiques pour réaliser les expériences décrites à la question 2. Donnez les résultats obtenus pour les expériences décrites à la question 2 sous un format approprié.

* Qualité évaluée : 5.2 Appliquer un outil d’ingénierie
  + Critère d’évaluation : Application réussie de programmes informatiques pour traiter des données et produire des résultats. Descriptions bien synthétisées des codes informatiques utilisés.

Text file : <https://www.mattmahoney.net/dc/textdata.html>

image :

* <https://paperswithcode.com/dataset/icb>
* <https://paperswithcode.com/datasets?mod=images>

## Question 4 (/8, 4 points par qualité)

Analysez les résultats obtenus et mettez-les en relation avec les hypothèses de la question 1. Est-ce que les hypothèses sont supportées par les résultats ?

* Qualité évaluée : 3.5 Analyser les résultats expérimentaux
  + Critère d’évaluation : Qualité et exhaustivité de l’analyse des résultats concernant leur cohérence avec le fonctionnement des méthodes appliquées.
* Qualité évaluée : 3.6 Vérifier les hypothèses et argumenter
  + Critère d’évaluation : Qualité et exhaustivité de l’analyse critique des résultats en fonction des hypothèses.