

УДК 004.627

*Козачок А. О., здобувачка 2 курсу спеціальності 122 Комп'ютерні науки,
Ветров О. С., старший викладач кафедри прикладної математики та
кібербезпеки*

СТИСНЕННЯ ДАНИХ ТА ЇХ ВПЛИВ НА ШВИДКОДІЮ АЛГОРИТМІВ ОБРОБКИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Зі збільшенням обсягів інформації, що обробляється в сучасному суспільстві, виникає потреба в оптимізації ефективного управління та обробки даних. Важливим аспектом цього завдання є методи стиснення даних, спрямовані на зменшення обсягу інформації без втрати її значимості.

Методи стиснення даних стали невід'ємною частиною розвитку цілої низки технологій – від зберігання і передачі даних до обчислень у реальному часі. Ці методи не лише економлять простір і ресурси, але й суттєво впливають на швидкість роботи алгоритмів, що є важливим параметром у сучасних обчислювальних середовищах [1].

Алгоритми стиснення зазвичай використовуються для зменшення розміру файла без видалення інформації. Це може збільшити їх ентропію та зробити файли більш випадковими, оскільки всі можливі байти стають більш загальними. Алгоритми стиснення також можуть бути корисними, коли вони використовуються для створення імітації через запуск функцій стиснення у зворотному порядку. Стиснення даних становить велику цікавість для всіх, хто хоче приховати дані з чотирьох причин: менше даних легше обробляти; стислі дані зазвичай більші; реверсивне стиснення може імітувати дані. Сьогодні використовується низка методів стиснення даних. За останні кілька років ця сфера надзвичайно розширилася через велику економічну цінність таких алгоритмів. Одними з найбільш популярних методів є ймовірнісні, словникові методи, кодування довжини серії, хвильові методи, фрактальні методи та адаптивні схеми стиснення. Усі ці схеми стиснення корисні в окремих доменах [2].

Поширені методи стиснення даних видаляють і замінюють повторювані елементи даних та символи. Елементи даних і символи видаляються та замінюються, щоб зменшити розмір даних. Дублюючі елементи (тобто надмірність), замінюючи ці послідовності коротшими значеннями способом заміни повторюваних елементів (тобто надлишкових) на коротші значення. Стиснення ненадлишкових даних (тобто дані без надлишковості, наприклад, випадкові сигнали або шум) не можна стиснути. Стиснення зашифрованої інформації також зазвичай неможливе.

Існує два типи стиснення даних.

1. Стиснення без втрат: оригінальні дані можна відновити без спотворень.
2. Стиснення зі втратами: стиснення під час якого дані можуть бути відновлені з невеликими (іноді значними) спотвореннями. Ці методи можна застосовувати тільки для таких типів даних, для яких втрата частини вмісту не приводить до суттєвого спотворення інформації. До таких типів даних належать відео- та аудіо-, графічні дані.

Існує багато різних практичних методів стиснення без втрати інформації, які, як правило, мають різну ефективність для різних типів даних та різних обсягів. Однак в основі цих методів лежать три теоретичних алгоритми [3]:

- RLE (Run Length Encoding) алгоритм;
- групи LZW алгоритми;
- Хаффмана алгоритми.

Алгоритм RLE (Run Length Encoding) – один з найстаріших і найпростіших алгоритмів стиснення графіків. Він розбиває зображення на байтовий ланцюжок уздовж растрових ліній. Простіше кажучи, алгоритм замінює однакову послідовність символів на відповідні пари «число-значення». Наприклад, рядок «AAAABBBBBGGG» перетворюється на «4A4B3G». Основне застосування цього алгоритму – для зображень з невеликою кількістю кольорів. Це стосується, наприклад, спеціальних або наукових зображень. Однак він не набув широкого поширення через свою простоту, швидкість і низьку ефективність, незважаючи на дуже низьке використання пам'яті під час операцій архівування та розпакування.

Алгоритм LZW названо на честь його розробників Лампела, Зів і Велча; на відміну від RLE, цей алгоритм базується на тому самому ланцюжку байтів. Алгоритм стиснення можна описати так. Спочатку послідовно зчитуються всі символи з вхідних даних і перевіряється таблиця рядків (яка буде згенерована) на предмет існування такого рядка. Якщо відповідний рядок знайдено, процес зчитування продовжується (символічно), якщо такого рядка не існує, до таблиці додається 13-й рядок і процес повторюється. Основними цільовими зображеннями цього алгоритму є 8-розрядні зображення. Варто зазначити високу гнучкість цього алгоритму – у поширених архіваторах використовується декілька його варіантів.

Алгоритм Хаффмана вважається одним із класичних алгоритмів, відомих давно (з 60-х рр. XX ст.). Він базується виключно на частоті появи одного і того ж байта на зображенні. Алгоритм присвоює коротші бітові ланцюжки символам, які найчастіше зустрічаються у вхідному потоці. І навпаки, довші ланцюжки призначаються символам, які зустрічаються доволі рідко. Алгоритм вимагає двох проходів над вихідним зображенням. Варто зазначити, що алгоритм Хаффмана не використовується як окремий алгоритм. Насправді він є частиною більш склад-

ної схеми алгоритму багатоступеневого стиснення. Прикладом цього є формат даних TIFF, де алгоритм Хаффмана є лише одним із декількох кроків [4].

Отже, ефективне управління та обробка даних є важливим питанням у сучасному суспільстві, де різноманітної інформації багато і її обсяг постійно зростає. Технологія стиснення даних є важливим інструментом для оптимізації обробки великих масивів інформації та зменшення обсягу інформації без втрати її важливості. Було розглянуто кілька методів стиснення, зокрема RLE, групи KWE та алгоритм Хаффмана, кожен з яких має свої переваги та обмеження залежно від застосування.

Список використаних джерел

1. Salomon D. Data Compression: The Complete Reference. Springer London. 2007. DOI: 10.1007/978-1-84628-603-2.
2. Compression Algorithm. URL: <https://www.sciencedirect.com/topics/computer-science/compression-algorithm> (дата звернення: 20.10.2023).
3. Основи інформаційних технологій. URL: <https://informatics6.webnode.com.ua/zanyattu-10/> (дата звернення: 20.10.2023).
4. Олашин О. О. Інтелектуальна система покращення алгоритмів стиску зображення. URL: <https://ela.kpi.ua/server/api/core/bitstreams/d2665c09-73e8-4319-8dc6-82bca5d989c8/content> (дата звернення: 20.10.2023).

УДК 004.056.5:004.65

*Коновалюк І. Л., здобувач 2 курсу спеціальності 122 Комп'ютерні науки,
Зелінська О. В., канд. техн. наук, доцент, в. о. завідувача кафедри інформаційних технологій*

ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ БАЗ ДАНИХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Кожен бізнес, корпорація чи державна установа має інформаційну базу, яка включає дані про клієнтів, нормативні акти, продукти та фінансову звітність. Великі об'єми інформації зазвичай містять конфіденційні, корпоративні та особисті дані. Несанкціонований доступ до цих даних може призвести до серйозних наслідків у фінансовій і репутаційній сферах.

Існують дві ключові причини, які зумовлюють необхідність для приватних компаній та державних установ вкладати значні ресурси у захист баз даних.

По-перше, це кіберзлочинність. Зловмисники постійно удосконалюють свої методи, використовують нові види програм-вимагачів та безфайлові методи