КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (24), 2024
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2024.24.266281
УДК 681.327

Курбет Павло Миколайович
аспірант
Інститут телекомунікацій та глобального
інформаційного простору НАН України, Київ, Україна
ORCID ID: 0000-0002-0612-3859
tovsba@gmail.com

МЕТОД ПІДГОТОВКИ НАЧАЛЬНИХ ПОЛІНОМІВ ДЛЯ РЕКУРСИВНИХ СИСТЕМАТИЧНИХ ЗГОРТОЧНИХ КОДІВ ТУРБО КОДІВ З ВИКОРИСТАННЯМ ГЕНЕТИЧНОГО АЛГОРИТМУ

Анотація. Стаття присвячена підвищенню ефективності функціонування безпроводових систем передачі інформації з адаптацією за рахунок підготовки начальних поліномів рекурсивних систематичних згорточних кодів турбо кодів з використанням генетичного алгоритму. В якості цільової функції запропонований показний кількості змін знаку апріорно-апостеріорної інформації декодера турбо коду для певної вибірки біт даних. В якості апріорної інформації використовується значення канальних символів з урахуванням функції «надійності» каналу, яка вказує на рівень дисперсії завад типу адитивний білий гаусівський шум. В якості апостеріорної інформації використовується логарифм відношення функцій правдоподібності про переданий біт даних. Аналіз відомих робіт показує, що при використанні адаптивних систем з кодуванням в якості параметра, що адаптується використовується швидкість кодування, яка регулюється кількістю перевірочних символів з виходу кодера турбо кода, при цьому відсутні розробки з адаптації поліномів турбо кодів, а також з швидкого формування начальних поліномів рекурсивних систематичних згорточних кодів турбо кодів. Використання раціональних поліномів в якості начальних при адаптації дозволить ефективніше використовувати енергетичну ефективність безпроводових систем передачі даних. Стаття складається із вступу, де висвітлено проблему, проведено аналіз останніх досліджень та публікацій по цій тематиці та сформульовано мету статті. Показано результати дослідження, зроблено висновки та перспективи подальших досліджень. Завершуються стаття списком використаних джерел. Як результат роботи запропонованого методу наведено первинні поліноми турбо кодів, які були знайдені із застосуванням генетичного алгоритму для каналу з адитивним білим гаусівським шумом. Напрямком подальший досліджень вважаємо пошук початкових перемежувачів між компонентними рекурсивними систематичними згорточними кодами турбо кодів.

Ключові слова: коригуючі коди; турбо коди; безпроводові системи передачі даних; функції правдоподібності; адаптація.

ВСТУП

Постановка проблеми. На сьогоднішній день в світі впроваджуються системи безпроводового доступу за технологією 5G.

В сучасних безпроводових системах передачі даних для підвищення достовірності інформації застосовуються технології адаптивного кодування, OFDM, MIMO.

Під складними завданнями оптимізації з обмеженнями розуміються такі, для яких відсутня апріорна інформація про цільову функцію Q, яка може бути використана для організації пошуку оптимального рішення, або складність одержання цієї інформації неприйнятна. Можна виділити наступні основні особливості складних завдань оптимізації:

1. Функція Q може задаватися правилами (алгоритмами) їхнього обчислення.

2. Негладкий характер функції Q.
3. Відсутність інформації про похідні функції Q або її похідні не є неперервними.

Для безпроводових систем із турбо кодами інформація про функцію Q та її похідну відсутня.

Всі розроблені на сьогоднішній день методи рішення задач оптимізації, залежно від необхідної якості одержання результату, можна віднести до одного із двох класів:

– методи, які завжди приводять до знаходження оптимального рішення, але вимагають для цього в найгіршому випадку неприпустимо великої кількості операцій;
– методи, які не завжди приводять до знаходження оптимального рішення, але вимагають для одержання цього рішення прийнятного числа операцій.

Розглянуті особливості не дозволяють вирішувати завдання оптимізації за допомогою чисельних методів, як завдання багатофакторного пошуку екстремуму, тому що воно відноситься до складних завдань оптимізації. Однак, з погляду на те, що область рішень є кінцевою множиною, завдання може бути вирішено перебором всіх можливих значень вектора з використанням методів і алгоритмів послідовного звуження множини рішень (повний і цілеспрямований перебір). У випадку використання складного рекурсивного систематичного згорточного коду (РСЗК) турбо коду, для формування первинних поліномів для ініціалізації начального стану кодера турбо коду при використанні в адаптивній системі, метод перебору буде неприйнятний у зв’язку із тривалими обчислювальними процесами. Тому для рішення завдання ефективного пошуку необхідно використовувати методи, що забезпечують більш швидке знаходження потрібного результату, чим метод повного перебору. До таких відносяться методи та алгоритми послідовного поліпшення рішень.

Ітеративні алгоритми мають високу збіжність, однак погрішність зі збільшенням вихідних даних збільшується.

У випадку використання випадкового або спрямованого випадкового пошуку як:

ГА є найбільш універсальними в порівнянні з іншими розглянутими алгоритмами, тому що швидкість збіжності цих алгоритмів висока та не залежить від ЦФ. Вони дозволяють знайти глобальний екстремум ЦФ за найменше число ітерацій [1] – [6].

ГА являє собою адаптивний пошуковий метод, що заснований на селекції кращих елементів у популяції.

Основою для виникнення генетичних алгоритмів послужили модель біологічної еволюції та методи випадкового пошуку.

Еволюційний пошук з погляду перетворення інформації — це послідовне перетворення однієї кінцевої множини проміжних рішень в іншу. Саме перетворення можна назвати алгоритмом пошуку або генетичним алгоритмом. ГА здійснюють пошук балансу між ефективністю і якістю рішень за рахунок «виживання найсильніших альтернативних рішень» у невизначених і нечітких умовах.

Аналіз останніх досліджень і публікацій. У статті [7] представлена нову структура турбо коду, яка підходить для паралельного та послідовного з’єднання для дуже низького BER, коли відношення сигнал-шум E_b/N_0 досягає певної точки в області мінімальної помилки. Пропонований код є модифікованим турбо кодом, який використовує паралельну та послідовну конкатенацію компонентних кодів. Порівняно з турбо-кодами, запропонований код може досягти майже такої ж складності декодування зі значно зниженою продуктивністю BER в області мінімальної помилки. Результати

моделювання показують, що запропонований турбо код забезпечує хорошу продуктивність. Пропонований турбо код легко адаптується до великого діапазону розмірів блоків даних.

У роботі [8] пропонується нова система адаптивної модуляції (AM) з LDPC-кодами з підтримкою машинного навчання (ML), де використовуються коди LDPC з короткою довжиною блоку. Звичайна система адаптивної модуляції та кодування (AMC) включає метод перегляду фіксованої таблиці, який також називають адаптацією зв'язку внутрішнього циклу (ILLA) та адаптацією зв'язку зовнішнього контуру (OLLA). Для ILLA адаптивна здатність досягається шляхом перемикання режимів модуляції та кодування на основі пошукової таблиці з використанням порогових значень співвідношення сигнал/шум (SNR) при цільовій частоті бітових помилок (BER), тоді як OLLA базується на методі ILLA шляхом динамічного регулювання порогів SNR для подальшої оптимізації продуктивності системи.

Метою статті є підвищення ефективності функціонування безпроводових систем передачі інформації за рахунок підготовки начальних поліномів рекурсивних систематичних згорточних кодів турбо кодів з використанням генетичного алгоритму.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Генетичний алгоритм — класичний алгоритм серед природних евристичних алгоритмів. Вперше його запропонував Джон Холланд з Мічиганського університету в 1960-х роках. Як випливає з назви, генетичний алгоритм — це алгоритм, розроблений і реалізований відповідно до процесу біологічної спадковості та еволюції в природі. З прогресом та розвитком науки і техніки продуктивність комп'ютера якісно покращилася, і генетичний алгоритм поступово вийшов у сферу практичного застосування з теоретичного рівня. Будучи каркасним алгоритмом, який може ефективно вирішувати складні задачі оптимізації, генетичний алгоритм набуває все більшого поширення завдяки поглибленим дослідженням вчених і широко використовується в багатьох галузях, таких як: обчислювальна техніка, торгівля, сільське господарство тощо.

Генетичний алгоритм вчиться на режимі еволюції природних організмів, алгоритмізує процес біологічної еволюції та моделює його на комп'ютері, щоб вирішити проблеми оптимізації в реальному полі. Це глобальний алгоритм пошуку, який може уникнути обмеження локальним максимумом. Він буде випадковим чином генерувати різні типи рішень проблем і вибирати більш сприятливі рішення відповідно до принципу виживання сильнішого. Цей алгоритм додатково повторюватиме та оптимізуватиме рішення за допомогою спадковості та варіацій, що подібно до біологічної еволюції в природі. Саме рандомізація його початкової схеми відбору та безперервні варіації в генетиці дозволяють алгоритму вийти з дилеми локальної оптимізації та стати придатним для глобального пошуку та оптимізації.

Генетичний алгоритм — це алгоритм, заснований на теорії еволюції Дарвіна. Він підходить для вирішення складних завдань. Також він має переваги самонавчання, самоадаптації та самооптимізації. Основний принцип генетичного алгоритму: він абстрактно кодує розв'язок задачі в хромосомах, і одна хромосома відповідає одному розв'язку. Кожну особину відбирають і оцінюють відповідно до придатності наступного покоління за допомогою генетичного алгоритму з вищою ймовірністю перехрещування та мутації, а потім кожну особину відбирають відповідно до потреб наступного

покоління. Остаточна особина — це те, що ми хочемо знайти в результаті оптимального розв'язку задачі.

У генетичному алгоритмі є три конкретні кроки: операція відбору, операція перехрещування та операція мутації. Лише шляхом наукового та обґрунтованого об'єднання цих операційних кроків і розробки потрібної схеми роботи відповідно до конкретних проблем, які потрібно вирішити, можна максимально покращити продуктивність алгоритму та швидко й точно отримати оптимальне рішення. Операція відбору стосується відбору особин із високою придатністю в поточній популяції для наступного етапу успадкування та варіації. Імовірність індивідуального відбору прямо пропорційна значенню придатності, тобто чим вище значення придатності, тим більша ймовірність відбору. Перехресна операція стосується випадкового поєднання батьків, вибраних у вищезазначеній операції відбору. Таким чином хромосоми кожної вибраної особини обмінюються з певною ймовірністю обміняти геном, відповідний хромосомі. Процес перехрещування полягає у виборі двох хромосом із популяції та випадковому виборі однієї чи кількох позицій хромосом для обміну. Загальне емпіричне значення знаходиться між 0,01 і 0,99. Операція мутації стосується вибору особини з популяції та вибору однієї або кількох точок у її хромосомі для мутації з метою створення кращих особин. Це засіб впровадження для створення нових індивідів. Використання операції мутацій забезпечує різноманітність популяції при відтворенні. Це може не тільки додатково оптимізувати ефективність генетичного алгоритму, але й змусити алгоритм шукати область поза поточним фокусом, щоб уникнути передчасної конвергенції. Його загальне емпіричне значення становить від 0,0001 до 0,1.

Для задач оптимізації з різними властивостями в різних галузях науковці запропонували різні типи алгоритмів інтелектуальної оптимізації, такі як алгоритм моделювання відпалу та алгоритм оптимізації рою частинок. Ці алгоритми базуються на різних теоріях, застосовні до різних конкретних галузей, і мають свої переваги та недоліки. Як алгоритм оптимізації, придатний для вирішення складних задач, алгоритм спадщини має такі характеристики:

1) Генетичний алгоритм — це стохастичний алгоритм оптимізації, який не має занадто багато математичних вимог до задачі оптимізації. З огляду на її еволюційні характеристики, внутрішні атрибути проблеми не потрібно враховувати в процесі пошуку. Лінійний чи нелінійний, дискретний чи безперервний, він може безпосередньо керувати структурним об'єктом і має широкий спектр сценаріїв і діапазонів застосування.

2) Генетичний алгоритм безпосередньо бере цільову функцію як пошукову інформацію, використовує лише функцію придатності для оцінки індивідуума без іншої складної похідної та додаткової інформації, та виконує генетичну операцію на цій основі для реалізації обміну інформацією між індивідами в популяції. Саме тому він менше залежить від вирішення проблеми та має добру гнучкість.

3) Генетичний алгоритм використовує багатоточковий паралельний метод пошуку, який не обмежується однією точкою, тому він може ефективно запобігти зближенню процесу пошуку до локального оптимального рішення. У той же час, завдяки характеристикам паралельного розрахунку генетичного алгоритму, ми можемо покращити швидкість роботи алгоритму за допомогою великомасштабних паралельних обчислень, щоб була можлива швидка оптимізація в реальному часі.

4) Генетичний алгоритм працює для кодування параметрів, а не для самих параметрів. Його правила оптимізації залежать від відповідної ймовірності, а не достовірності. По суті, це не тільки алгоритм оптимізації, але й гарантія загальної основи для вирішення задач оптимізації системи, яка має великий розвиток.

ISSN 2663 - 4023

Загальна швидкість кодування турбо коду $R = 1/3$. Кожен РСЗК представляється поліноміальними генераторами виду [9] – [10]: $G(D) = \left[1, \frac{g_1(D)}{g_0(D)}\right]$, де, наприклад, $g_1(D) = 1 + D + D^3$ — поліном прямого зв'язку РСЗК (їх може бути декілька), $g_0(D) = 1 + D^2 + D^3$ — поліном зворотного зв'язку РСЗК.

На рис. 1 показана схема РСЗК турбо коду виду (1, 47/43, 53/43, 33/43) з кодовим обмеженням $K = 6$ із швидкістю кодування $R = 1/4$. В цьому випадку структура має вигляд:

$$1,\ g_1 / g_0, g_2 / g_0, g_3 / g_0,$$

де g_0 — поліном зворотного зв'язку, а g_1 — перший поліном прямого зв'язку, g_2 — другий поліном прямого зв'язку, g_3 — третій поліном прямого зв'язку.

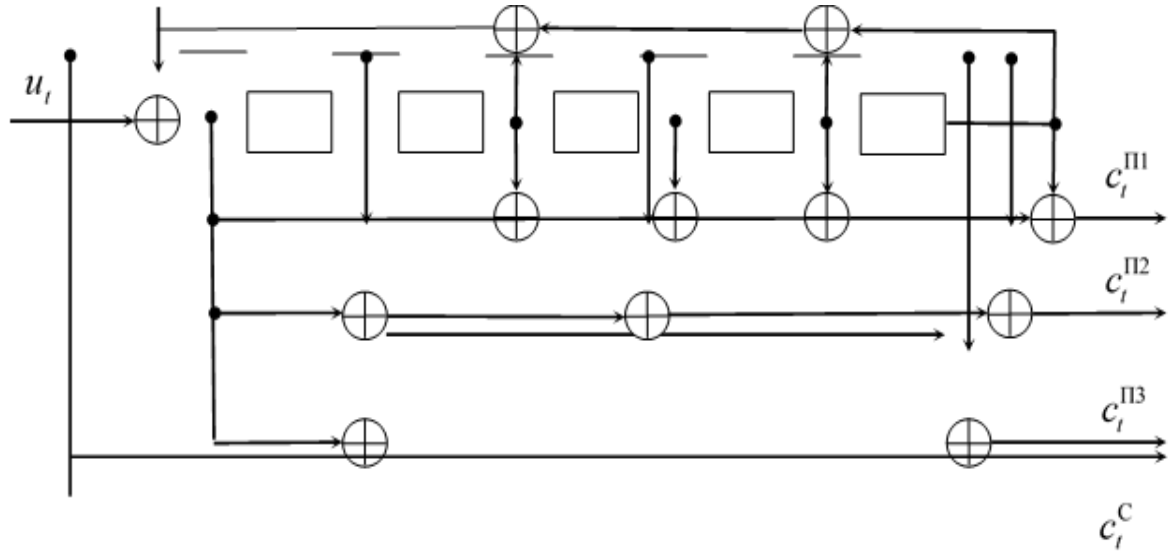

Рис. 1. Схема рекурсивного систематичного згорточного коду при K = 6 виду (1, 47/43, 53/43, 33/43)

Відповідно для цього РСЗК варіант побудови кодера турбо кода буде мати наступний вигляд (рис. 2), де П — пристрій перемеження біт даних.

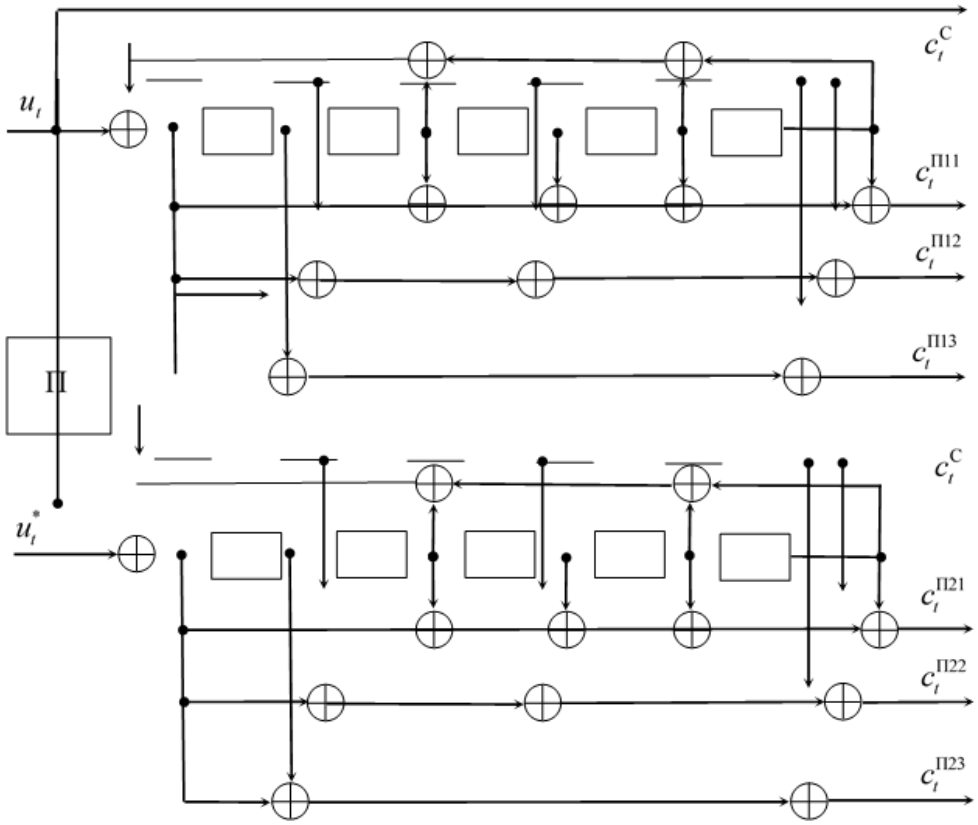

Рис. 2. Схема кодера турбо кода при використанні двох РСЗК K = 6 виду (1, 47/43, 53/43, 33/43)

Розглянуті варіанти схем побудови РСЗК використовуються в кодерах турбо кода, а відповідні до кожного кодера решітчасті діаграми — в декодерах при декодуванні за максимумом апостеріорної ймовірності, використовуючи алгоритм BCJR або його модифікації.

РСЗК виконує кодування біт даних, використовуючи відповідну гратчасу діаграму, яка описується поліноміальними генераторами. Із збільшенням порядку полінома збільшується розмір гратчастої діаграми як 2^M (M — кількість елементів пам'яті РСЗК).

ГА відрізняються від інших оптимізаційних і пошукових процедур наступним:

1. Працюють із закодованою множиною параметрів.
2. Здійснюють пошук не шляхом поліпшення одного рішення, а шляхом використання відразу декількох альтернатив на заданій множині рішень.
3. Використовують ЦФ, а не її різні збільшення для оцінки якості прийняття рішень.
4. Застосовують не детерміновані, а імовірнісні правила аналізу оптимізаційних завдань.

Для формалізації опису моделі ми пропонуємо наступну формалізацію: задано n елементів, $X_1, X_2, ..., X_n$, де кожен елемент X_i є вектором, описаним m ознаками $X_i = (x_{i,1}, x_{i,2}, ..., x_{i,m})$, та задано ідеальний оптимальний розв'язок $\overline{X} = \{\overline{x}_1, \overline{x}_2 ..., \overline{x}_m\}$. Можливо отримати змішання X_{mix} елементів, яке було б якомога ближче до «ідеального

оптимального розв'язку». Змішання отримують як конвексну комбінацію *n* елементів, тобто:

$$X_{mix} = a_1 X_1 + a_2 X_2, ..., a_n X_n,$$
$$a_i \geq 0, \sum_{i=1}^{n} a_i = 1 \quad (1)$$

Проблема полягає у тому, щоб знайти $a_i \ \ i = 1,....n$, такі, щоб відстань $d(\overline{X}, X_{\min})$ була якомога меншою.

Для вирішення проблеми запропоновано наступне кодування: Позначимо хромосому як *X*, батька – *P*, і дитину – *C*. Застосовується подвійне кодування таким чином, що кожен *X*, *P* або *C* завжди представлений двома векторами: бінарним вектором, що представляє вибір ТАК/НІ кожного елемента (бінарне кодування вектора), та вектором відсотків, який показує взяті відсоткові значення кожного елемента (пряме кодування вектора), обидва розмірності n:

$$x_b = \{x_{b,1}, x_{b,2}, ..., x_{b,i}, ..., x_{b,n}\} \quad (2)$$

$$x_p = \{x_{p,1}, x_{p,2}, ..., x_{p,i}, ..., x_{p,n}\} \quad (3)$$

X_b показує бінарну хромосому з $x_{b,1} = 1$, що представляє, чи обрано i-й елемент ($x_{p,i} = 0$ в іншому випадку), X_p показує відсоткову хромосому, де $x_{p,i} \in [0,1]$, з $\sum_{i=1}^{n} x_{p,i} = 1$.

Генетичний алгоритм поділяється на дві основні частини: перше покоління і наступні, кожна складається з набору послідовних дій. У першій ітерації послідовність кроків виглядає наступним чином:

1) Покоління батьків: генерується число d_{start} бінарних хромосом X_b, після чого для N_{r_p} разів створюються взаємодоповнюючі випадкові відсотки, утворюючи до $d_{start} \cdot X_{r_p}$ відсоткових хромосом X_p.

Для генерації кожної відсоткової хромосоми X_p, нехай $k \leq n$ позначає кількість обраних елементів. Де $x_{b,i} = 0$, навіть $x_{p,i} = 0$, в той час як у випадку $x_{b,s} = 1$, нехай $1 \leq s \leq k$ вказує на обраний елемент s-го розглянутого для генерації відсоткового значення, а $rand(\min; \max)$ — це функція, яка генерує випадкове значення між мінімальним і максимальним краями виключно, кожне значення X_p обчислюється як:

ISSN 2663 - 4023

$$x_{p,s}\begin{cases} x_{p,s}; & k=1, s=1 \\ rand(0;1) & k\geq 2, s=1 \\ rand(0;1-\sum_{i=1}^{s-1} x_{p,i}) & k\geq 2,\ 2\leq s\leq k \\ 1-\sum_{i=1}^{s-1} x_{p,i} & k\geq 2, s=k \end{cases} \quad (4)$$

2) Покоління дітей: кожну відсоткову хромосому X_p досліджують N_{Expl_p} разів, випадковим чином переміщуючи точки відсотків у фіксованому сусідньому середовищі початкових відсоткових значень, забезпечуючи, що загальна сума нових відсоткових значень дорівнює одиниці (5). Таким чином, генеруються до $d_{start} \cdot N_{rp} \cdot N_{Expl_p}$ відсоткових дітей C_p.

$$x_p = \{x_{p,1}+\Delta x_{p,1},...,x_{p,n}+\Delta x_{p,i},...,x_{p,i}+\Delta x_{p,n}\},\ \sum_{i=1}^{n} x_{p,1}+\Delta x_{p,i}=1 \quad (5)$$

Кількість елементів, які обрано змінювати кожного разу, є випадковою, і для проведення дослідження потрібні як бінарні, так і відсоткові хромосоми. Коли обрано один елемент ($k=1$), дослідження не виконується. Коли обрано два елементи ($k=2$), перший змінюється на $\Delta x_{p,1}\in[-\alpha;+\alpha]$, а другий на $-\Delta x_{p,1}$. Нарешті, коли обрано три або більше елементів ($k\geq 3$), застосовується рекурсивна формула. Весь процес дослідження відсотків може бути представлений через наступні рівняння.

Перше змінення відсотку задається як:

$$\Delta x_{p,1} = rand(-\alpha;+\alpha) \quad (6)$$

Від другого до передостаннього $(2\leq s\leq k)$:

$$\Delta x_{p,s} = rand(-L;+U) \text{ with } \begin{cases} U=+\alpha-\sum_i \Delta x_{p,i} & \forall \Delta x_{p,i}>0 \\ L=-\alpha+\sum\left|\Delta x_{p,i}\right| & \forall \Delta x_{p,i}<0 \end{cases} \quad (7)$$

Потім, останнє змінення $(s=k)$:

$$\Delta x_{p,k} = -(\sum_{i=1}^{k-1} \Delta x_{p,i}) \quad (8)$$

ПРИМІТКА: кожного разу, коли генерується нове значення зміни відсотку $\Delta x_{p,i}$, алгоритм забезпечує, що виконується наступна умова:

$$\Delta x_{p,i}+x_{p,i}>0, \text{ та } \Delta x_{p,i}+x_{p,i}<1, \quad (9)$$

3) Оцінка та відбір: в кінці ітерації буде доступний набір пар бінарних та відсоткових хромосом (C_b, C_p). Серед них потрібно вибрати кращі *d*. Спочатку, за допомогою визначеної метрики, потрібно порівняти всі *C*, щоб досягти такої мети. Потім за допомогою методу відбору потрібно обрати найкращі.

ISSN 2663 - 4023

Щоб порівняти дітей *C* між собою, першим кроком є отримання кількостей кожного елемента вздовж окремих ознак та їх сумування по всім ознакам. Таким чином, обчислюються значення суміші $x_{\min}$ для кожного *C* та можуть порівнюватися зі значеннями бажаного результату. Цілу процедуру можна узагальнити як операцію матричного множення типу:

$$X^T \cdot X_p = X_{mix} \tag{10}$$

$$\begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,j} & \dots & x_{1,m} \\ x_{2,1} & x_{2,2} & \dots & x_{2,j} & \dots & x_{2,m} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{i,1} & x_{i,2} & \dots & x_{i,j} & \dots & x_{i,m} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n,1} & x_{n,2} & \dots & x_{n,j} & \dots & x_{n,m} \end{pmatrix}^T \cdot \begin{bmatrix} x_{1,1} \\ x_{p,2} \\ \dots \\ x_{p,i} \\ \dots \\ x_{p,n} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_{i,1} \cdot x_{p,i} \\ \sum_{i=1}^{n} x_{i,2} \cdot x_{p,i} \\ \dots \\ \sum_{i=1}^{n} x_{i,j} \cdot x_{p,i} \\ \dots \\ \sum_{i=1}^{n} x_{i,m} \cdot x_{p,i} \end{bmatrix} = \begin{bmatrix} x_{mix,1} \\ x_{mix,2} \\ \dots \\ x_{mix,i1} \\ \dots \\ x_{mix,n} \end{bmatrix} \tag{11}$$

У 11 кожний елемент матриці X_{mix} представляє значення суміші ознак.

Після цього кроку значення X_{mix} можна порівняти зі значеннями бажаної оптимальної суміші $\overline{X}$, використовуючи об'єктивну функцію мінімізації, яка прямує до нуля, якщо суміш X_{mix} близька до бажаного кінцевого елемента $\overline{X}$.

Для роботи ГА вибирають множину параметрів оптимізаційного завдання та кодують їх у послідовність кінцевої довжини в деякому алфавіті. Вони працюють доти, поки не буде виконано задане число генерацій (ітерацій алгоритму) або на деякій генерації буде отримане рішення певної якості, або, коли знайдений локальний оптимум, тобто виникла передчасна збіжність і алгоритм не може вийти із цього стану. На відміну від інших методів оптимізації ці алгоритми, як правило, аналізують різні області простору рішень одночасно і тому вони більш пристосовані до знаходження нових областей із кращими значеннями цільової функції (ЦФ).

Функціонування ГА схематично можна представити таким чином (рис. 3):

1. Згенерувати випадковим чином популяцію розміру *V*.
2. Обчислити ЦФ для кожного рядка популяції.
3. Виконати операцію селекції (етап 1).
4. Виконати операцію схрещування (етап 2).
5. Виконати операцію мутації (етап 3).
6. Якщо критерій останова не досягнутий, то перейти до кроку 2, інакше — завершити роботу.

Популяція — це множина бітових рядків, кожний з яких представляє в закодованому вигляді одне з можливих рішень завдання. Для кожного рядка обчислюється ЦФ, що характеризує якість рішення. Як початкова популяція може бути використаний довільний набір рядків. Основні операції алгоритму: етап 1, етап 2 та етап 3 — виконуються над елементами популяції. Результатом їхнього виконання є чергова популяція. Даний процес триває ітераційно доти, поки не буде досягнутий критерій зупинення.

Основним питанням при формулюванні завдання оптимізації є вибір ЦФ Q, що дозволить кількісно оцінити ефективність роботи системи.

Для пошуку начальних поліномів РСЗК турбо коду при впливі завад передбачається, що вектори сигнальної структури визначені, тобто $\vec{F}_i^* \in E^*_F$, $i \in \overline{1,n}$, де n — розмір отриманої підмножини.

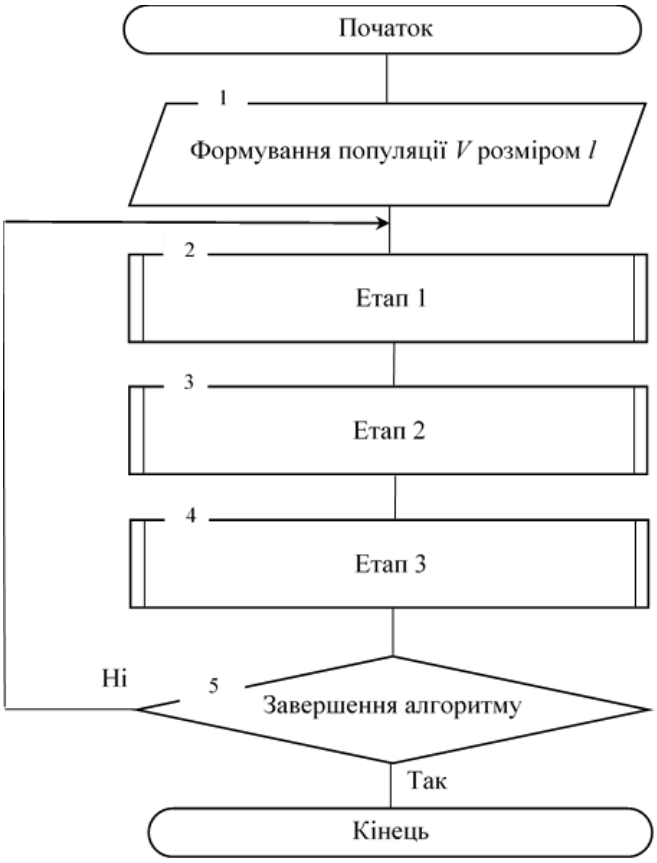

Рис. 3. Схема функціонування генетичного алгоритму

Тому при начальних поліномах РСЗК турбо коду будуть використовуватись елементи $g_1, g_2, \ldots, g_h$. Для зручності представимо їх у вигляді вектора $G^{/} = \{g_1, g_2, \ldots, g_h\}$.

Кожний бітовий рядок використовується для кодування вихідної множини альтернатив і являє собою впорядковані набори з d елементів: $\vec{G}^{/} = \left(g_1^{/}, g_2^{/} \ldots g_d^{/}\right)$, кожний з яких представлений у двійковому алфавіті. Множина подібних рядків утворює множину рішень (популяцію) : $V = \left(\vec{G}_1^{/}, \vec{G}_2^{//} \ldots \vec{G}_l^{//}\right)$, де l — розмір популяції, кожне з яких може бути одним з можливих варіантів конфігурації ТК.

ISSN 2663 - 4023

Для кожного рядка популяції обчислюється ЦФ за правилом: $Q_i = \frac{\sum_{h=1}^{N^*} \Phi^*_{ih}}{N^*}$, $i \in \overline{1,l}$, де N^* — кількість переданих блоків символів, що володіють однаковою конфігурацією вектора $\vec{G}^{/}$, Φ^*_i — нормований показник декодування прийнятої послідовності. З метою спрощення визначення показника декодування, одержимо нормовану величину Φ^* таким чином:

$$\Phi^* = \frac{\Phi - \Phi^{(-)}}{\Phi^{(+)} - \Phi^{(-)}} \cdot 100\% = \frac{\Phi - N}{N\left(I - 0{,}5\right)} \cdot 100\% \quad (12)$$

Для величини Φ^* значення 100 % відповідає абсолютній неефективності, а 0 % — абсолютній ефективності ітеративного декодера при обробці інформаційного блоку.

Використання N^* переданих блоків у виразі для розрахунку ЦФ обумовлене тим, що Φ^* є випадковою величиною. Необхідна кількість блоків N^* обрана відповідно до функції Лапласа.

Величина Φ може бути визначена як сума кількостей змін знаку апріорної-апостеріорної інформації ψ_{kj} про переданий символ для всього блоку розміром N по всіх декодерах всіх ітерацій декодування:

$$\Phi = \sum_{j=1}^{I} \sum_{k=1}^{2} \psi_{kj}, \quad (13)$$

де I — кількість ітерацій декодування.

Сутність величини Φ полягає в тому, що вона характеризує правдоподібність даних, отриманих при декодуванні інформаційного блоку. Чим менше Φ, тим достовірніше був декодований прийнятий інформаційний блок.

У роботі ітеративного декодера ТК можливі два крайніх випадки:

1. Канальні завади не впливають на передану інформацію або вплив на стільки малий, що ними можна знехтувати.
2. Канальні завади впливають на передану інформаційну послідовність так, що коректне декодування неможливе.

У першому випадку величина Φ прийме своє найменше значення, що чисельно буде рівнятися кількості переданих інформаційних біт:

$$\Phi^{(-)} = N.$$

Це відповідає випадку, коли прийнята послідовність буде успішно декодована на першому декодері першої ітерації.

У другому випадку величина Φ прийме своє найбільше значення, що може бути визначене як:

$$\Phi^{(+)} = \frac{N}{2}\left(2I + 1\right).$$

Значення показника декодування Φ для реального процесу ітеративного декодування будуть відповідати інтервалу між двома граничними значеннями: $\Phi^{(-)} \le \Phi \le \Phi^{(+)}$.

ISSN 2663 - 4023

Структура РСЗК турбо кода записується в наступному виді: ($1, g_1/g_0, \ldots, g_{n-1}/g_0$), де g_0 — поліном зворотнього зв'язку, а $g_1, \ldots, g_{n-1}$ — поліноми прямих зв'язків. Перетворення полінома в бітовий рядок відбувається таким чином: на перше місце ставиться цифра 1, записуються бітові комбінації, відповідальні за формування перевірочних символів, кількість яких визначається значенням *v*, потім — бітова комбінація, що формує зворотний зв'язок. Наприклад, при $K = 3$ і $v = 2$ для полінома [1,1011/1101,1001/1101] його бітове подання буде $\vec{G}'$ =1101110011101. Множина подібних рядків утворює популяцію. Кожний рядок є одним з можливих рішень завдання.

Традиційні ітеративні алгоритми займаються поліпшенням єдиного проміжного рішення, що повинне наблизитись до точного. Ітеративний ГА у відмінності від них працює в інтересах поліпшення всієї популяції в цілому, а не деякого конкретного значення вектора $\vec{G}'$, саме це необхідно при начальній ініціалізації РСЗК для швидкості прийняття рішення. У рамках процедури генетичного пошуку діють наступні обмеження:

1. Розмір популяції повинен залишатися постійним.
2. Довжина бітових рядків популяції повинна залишатися незмінною.
3. Критерієм зупинки служить виконання заздалегідь зазначеної кількості ітерацій.

Етап 1 служить для модифікації популяції на черговій ітерації алгоритму, що полягає у видаленні невдалих рядків і розмноженні рядків із кращим значенням ЦФ. Рядок зі значенням цільової функції Q_i дасть у нову популяцію $\frac{Q_i}{\overline{Q}}$ рядків, де $\overline{Q}$ — середнє значення ЦФ у поточній популяції.

Таким чином, запропонований метод представимо в наступному вигляді.

1. Перший член ряду дорівнює величині популяції, цілочисельно діленої на 2.
2. Кожний наступний член дорівнює попередньому, діленому на 2 і округленому до більшого.

Наприклад, число 25 буде породжувати ряд: 25 (12 6 3 2 11)

Це значить, що в популяції з 25 рядками найбільш вдалий рядок дасть 12 рядків. Рядок, що зайняв друге місце в загальному рейтингу — 6 рядків. Третє — 3. Четверте — 2. П'яте та шосте — 1. Метою етапу 1 є одержання нової популяції того ж розміру з найбільш удалих рядків. Рядки в нову популяцію додаються послідовно, тобто в новій популяції вони будуть упорядковані по рейтингу, утворюючи при цьому блоки однакових рядків. Цей факт буде використаний на етапі 2.

Операція етапу 2 полягає у виконанні наступних дій:

1. Вибрати пари рядків.
2. Для кожної обраної пари із заданою ймовірністю виконати операції етапу 2, одержати два рядки і зробити в популяції заміну на ці рядки.

Операція має єдиний параметр — імовірність схрещування (p_c). Найпростіший варіант операції (однокрапкове схрещування) виконується таким способом:

1. Популяція розбивається на пари.
2. Для кожної пари випадковим образом генерується число p'_c, якщо $p'_c < p_c$, то вибирається випадкове ціле число *i* в інтервалі [1, *d*-1], де *d* — довжина

рядка, і рядки обмінюються фрагментами, що перебувають після *i*-го біта, у противному випадку нічого не відбувається.

Пари для етапу 2 генетичного алгоритму вибираються таким чином: перша взаємодіє з останньою, друга з передостанньою й т.д.

Таким чином, кількість пар, які можуть бути використані в популяції, завжди відома та дорівнює половині популяції. Якщо популяція має непарну кількість рядків, то центральний рядок ігнорується. Таким чином, всі рядки популяції можуть бути використані на етапі 2. Рядки розташовані в популяції так, що не буде відбуватися взаємодія однакових рядків, тому що перша половина популяції буде містити ідентичні рядки з найкращим рейтингом (найменшим значенням ЦФ).

Операція етапу 3 полягає в інвертуванні кожного біта кожного рядка популяції із заданою ймовірністю**.** Параметр операції — імовірність мутації (p_m). Операція виконується таким способом:

1. Для кожного біта генерується випадкове число p'_m.
2. Якщо $p'_m < p'_m$, то біт інвертується.

ГА не гарантує єдине оптимальне рішення, а видає кілька близько-оптимальних результатів, то єдине рішення серед всіх можливих варіантів може бути визначене шляхом імітаційного моделювання.

В табл. 1 представлені начальні поліноми, які були знайдені з використанням генетичного алгоритму при моделюванні модему ФМ-2, кодеку турбо коду із псевдовипадковим перемежувачем/деперемежувачем, $N = 1000$, алгоритмом декодування *Log Map* з п’ятиступінчатою апроксимацією, 8 ітерацій декодування, моделі дискретно-неперервного каналу зв’язку із флуктуаційним шумом.

Таблиця 1

Начальні РСЗК при використанні турбо коду з псевдовипадковим перемежувачем, *N* = 1000, алгоритмом декодування *Log Map* для різних швидкостей кодування РСЗК *R* при впливі АБГШ

	Довжина кодового обмеження			
	$K = 3$	$K = 4$	$K = 5$	$K = 6$
$R = 1/2$ (для ТК $R = 1/3$)	(1,7/5)	(1,15/17)	(1,23/35)	(1,65/57)
	(1,5/7)	(1,17/15)	(1,23/33)	(1,45/67)
				(1,73/77)
				(1,53/55)
				(1,71/41)
				(1,53/41)
$R = 1/3$ (для ТК $R = 1/5$)	(1,7/5,7/5)	(1,13/17,15/17)	(1,25/37,33/37)	(1,47/53,75/53)
	(1,5/7,5/7)	(1,11/17,13/17)	(1,25/33,37/33)	(1,53/47,75/47)
		(1,13/11,17/11)	(1,27/21,33/21)	(1,45/41,55/41)
		(1,11/15,17/15)	(1,35/21,27/21)	(1,43/47,71/47)
		(1,13/17,11/17)	(1,27/25,31/25)	(1,63/43,45/43)
		(1,17/11,13/11)	(1,37/21,31/21)	(1,71/67,53/67)
		(1,11/15,13/15)	(1,31/27,35/27)	(1,53/67,41/67)
			(1,35/33,25/33)	(1,43/41,71/41)
			(1,25/31,23/31)	
			(1,37/31,25/31)	

Таким чином, отримані начальні поліноми турбо кодів для забезпечення заданої цільової функції (невизначеності декодування), що дозволить зменшити кількість помилок, тим самим підвищить ефективність функціонування безпроводових систем передачі інформації.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

1. Стаття присвячена підвищенню ефективності функціонування безпроводових систем передачі інформації з адаптацією за рахунок підготовки начальних поліномів рекурсивних систематичних згорточних кодів турбо кодів з використанням генетичного алгоритму.
2. В якості цільової функції запропонований показний кількості змін знаку апріорно-апостеріорної інформації декодера турбо коду для певної вибірки біт даних.
3. Як результат роботи запропонованого методу наведено первинні поліноми турбо кодів, які були знайдені із застосуванням генетичного алгоритму для каналу з адитивним білим гаусівським шумом.
4. Напрямком подальший досліджень вважаємо пошук початкових перемежувачів між компонентними рекурсивними систематичними згорточними кодами турбо кодів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mirjalili, S. (2019). Genetic algorithm, *Evolutionary algorithms and neural networks*. 43–55. https://doi.org/10.1007/978-3-319-93025-1_4
2. Kramer, O. (2017). Genetic algorithm essentials. *Springer, 679.* https://doi.org/10.1007/978-3-319-52156-5
3. Hariyadi, P. M., Nguyen, P. T., Iswanto, I., & Sudrajat, D. (2020). Traveling Salesman Problem Solution using Genetic Algorithm. *J. Critical Reviews, 7(1),* 56–61. http://dx.doi.org/10.22159/jcr.07.01.10.10
4. Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications, 80.* 8091–8126. https://doi.org/10.1007/s11042-020-10139-6
5. Shrivastava, P., Dhingra, S.L. & Gundaliya, P.J. (2010). Application of genetic algorithm for scheduling and schedule coordination problems. *Journal of Advanced Transportation, 36(1),* 23–41. https://doi.org/10.1002/atr.5670360103
6. Nugroho, E. D., Wibowo, M. E. & Pulungan, R. (2017). Parallel implementation of genetic algorithm for searching optimal parameters of artificial neural networks. *3rd International Conference on Science and Technology-Computer (ICST),* 136–141. https://doi.org/10.36227/techrxiv.12657173.v1
7. Mensouri, M., & Aaround, A. (2016). Adaptive encoding/decoding for turbo codes. *International Conference on Big Data and Advanced Wireless Technologies (BDAW`16),* 1–7. https://doi.org/10.1145/3010089.3010096
8. Xie, C., El-Hajjar, M., & Xin Ng, S. (2023). Machine learning assisted adaptive LDPC coded system design and analysis. *IET Communications, 18,* 1–10. https://doi.org/10.1049/cmu2.12707
9. Berrou, C., Glavieux, A., & Thitimajshima, P. (1993). Near Shannon limit error-correcting coding and decoding: Turbo-codes. *International Conference on Communications (ICC-93), 1064*–1070. https://doi.org/10.1109/ICC.1993.397441

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (24), 2024
ISSN 2663 - 4023

Pavlo Kurbet
Postgraduate
Institute of Telecommunications and Global Information
Space of the National Academy of Sciences of Ukraine, Kyiv, Ukraine
ORCID ID: 0000-0002-0612-3859
tovsba@gmail.com

A METHOD FOR PREPARING INITIAL POLYNOMIALS FOR RECURSIVE SYSTEMATIC CONVOLUTIONAL TURBO CODES USING A GENETIC ALGORITHM

Abstract. The article is devoted to increasing the efficiency of wireless information transmission systems with adaptation due to the preparation of initial polynomials of recursive systematic convolutional codes of turbo codes using a genetic algorithm. As an objective function, an indicative number of sign changes of the a priori-posterior information of the turbo code decoder for a certain sample of data bits is proposed. As prior information, the value of channel symbols is used, taking into account the channel "reliability" function, which indicates the level of dispersion of additive white gaussian noise. The logarithm of the ratio of the likelihood functions about the transmitted bit of data is used as posterior information. The analysis of known works shows that when using adaptive systems with coding as an adaptable parameter, the coding speed is used, which is regulated by the number of check symbols from the output of the turbo code encoder, while there are no developments on the adaptation of turbo code polynomials, as well as on the rapid formation of initial polynomials recursive systematic convolutional codes turbo codes. The use of rational polynomials as initial ones during adaptation will allow more effective use of the energy efficiency of wireless data transmission systems. The article consists of an introduction, which highlights the problem, analyzes the latest research and publications on this topic, and formulates the purpose of the article. The results of the research are shown, conclusions and prospects for further research are drawn. The article ends with a list of used sources. As a result of the work of the proposed method, the primary polynomials of turbo codes, which were found using a genetic algorithm for a channel with additive white Gaussian noise, are given. We consider the search for initial interleavers between component recursive systematic convolutional codes of turbo codes to be the direction of further research.

Keywords: corrective codes; turbo codes; wireless data transmission systems; likelihood functions; adaptation.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Mirjalili, S. (2019). Genetic algorithm, *Evolutionary algorithms and neural networks*. 43–55. https://doi.org/10.1007/978-3-319-93025-1_4
2. Kramer, O. (2017). Genetic algorithm essentials. *Springer, 679.* https://doi.org/10.1007/978-3-319-52156-5
3. Hariyadi, P. M., Nguyen, P. T., Iswanto, I., & Sudrajat, D. (2020). Traveling Salesman Problem Solution using Genetic Algorithm. *J. Critical Reviews, 7(1),* 56–61. http://dx.doi.org/10.22159/jcr.07.01.10.10
4. Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications, 80.* 8091–8126. https://doi.org/10.1007/s11042-020-10139-6
5. Shrivastava, P., Dhingra, S.L. & Gundaliya, P.J. (2010). Application of genetic algorithm for scheduling and schedule coordination problems. *Journal of Advanced Transportation, 36(1),* 23–41. https://doi.org/10.1002/atr.5670360103
6. Nugroho, E. D., Wibowo, M. E. & Pulungan, R. (2017). Parallel implementation of genetic algorithm for searching optimal parameters of artificial neural networks. *3rd International Conference on Science and Technology-Computer (ICST),* 136–141. https://doi.org/10.36227/techrxiv.12657173.v1
7. Mensouri, M., & Aaround, A. (2016). Adaptive encoding/decoding for turbo codes. *International Conference on Big Data and Advanced Wireless Technologies (BDAW`16),* 1–7. https://doi.org/10.1145/3010089.3010096

8. Xie, C., El-Hajjar, M., & Xin Ng, S. (2023). Machine learning assisted adaptive LDPC coded system design and analysis. *IET Communications, 18*, 1–10. https://doi.org/10.1049/cmu2.12707
9. Berrou, C., Glavieux, A., & Thitimajshima, P. (1993). Near Shannon limit error-correcting coding and decoding: Turbo-codes. *International Conference on Communications (ICC-93), 1064*–1070. https://doi.org/10.1109/ICC.1993.397441