**Алгоритм Particle Swarm Optimization.**

Автори реалізації алгоритму, що розглядається в описі: MUSHEER AHMAD, ISHFAQ AHMAD KHAJA, ABDULLAH BAZ, HOSAM ALHAKAMI, AND WAJDI ALHAKAMI.

Метод рою часток - це евристичний метод глобальної оптимізації, який реалізовано на основі популяцій. Запропоновано даний метод Кеннеді та Еберхартом у 1995 році. PSO базується на уявленні про «ройовий» інтелект (поведінка зграї птахів, риб, комах, тощо). Птахи або розлітаються, або навпаки летять разом під час пошуку їжі, перш ніж вони знайдуть гарне місце, де ця їжа знаходитиметься. Під час пошуку їжі завжди є один птах, який відчуває та шукає їжу краще за інших, тобто такий птах з більшою ймовірністю знайде місце, де може знаходитися їжа, а це в свою чергу означає, що він матиме кращу інформацію про джерело їжі ніж інші. Оскільки птахи в процесі пошуку постійно діляться такою «гарною» інформацію, зграя зрештою переміститься до «кращого» місця, де з більшою ймовірністю можна знайти їжу. У PSO пересування «птахів», починається з одного місцезнаходження на інше, еквівалентно зграї, хороша інформація – еквівалентна найбільш оптимістичному рішенню, а передбачувана їжа еквівалентна найбільш оптимістичному рішенню за весь час роботи алгоритму.

Даний метод допомагає знайти найбільш оптимістичне рішення саме завдяки співпраці кожного елементу популяції, «птиці». Завдяки алгоритму можуть бути вирішені доволі складні проблеми оптимізації. Переваги дають змогу застосовувати його до багатьох областей оптимізації окремо та в поєднанні з іншими існуючими алгоритмами. Алгоритм використовується в таких галузях, як навчання нейронних мереж, оптимізація певних функцій, машинне навчання, обробка сигналів, тощо.

**Опис алгоритму Particle Swarm Optimization**

У алгоритмі Particle Swarm Optimization популяція складається з «N» часток, а розташування кожної з часток відповідає потенційному рішенню в d-розмірному просторі. На положення кожної частинки у рої впливає як найбільш оптимістична позиція під час її руху (індивідуальний досвід, що називається особистим кращим чи  частинки), так і положення найбільш оптимальної частинки в її сусідстві (близький досвід, що називається найкращим серед усіх або). Частинки зграї летять у пошуковий простір завдяки своїм можливостям розвідки та експлуатації та використовують найкращі особисті та найкращі позиції у світі, щоб досягти найкращого рішення в PSO. Крім того, кожна частинка характеризується швидкістю. Швидкість і положення кожної частинки переглядаються після кожної наступної ітерації алгоритму. Швидкість і положення кожної частинки визначаються на кожному кроці як:





де,  та  це швидкість та позиція частки «» на її «» кроці відповідно та -розмірне значення її позиції.  представляє собою -розмірне значення кожного «» елемента у його найбільш оптимістичній позиції.  це -розмірне значення найбільш оптимістичної позиції для всього «рою». Параметри  генеруються випадковим чином у межах .

Для вирішення задачі генерації S-Box’ів з гарними показниками надійності було обрано саме такий метод з наступних причин:

1. PSO може бути застосований як для наукового, так і для інженерного використання, оскільки він базується на певному інтелекті.
2. Обчислення, що виконуються в PSO є доволі нескладними.
3. Метод передбачає зменшення кількості параметрів для налаштування та прийняття обмежень у порівнянні з іншими методами оптимізації.

**Параметри захищеності S-Box’ів**

Алгоритм, описаний у статті, що розглядається, передбачає порівняння захищеності різних згенерованих S-Box’ів на основі розрахунку та порівняння їхньої нелінійності, а саме: чим більшою є нелінійність блоку, тим більш надійним та захищеним він є. Це не завжди є правильним твердженням, оскільки на захищеність та надійність S-Box’ів впливає безліч параметрів, таких як: нелінійність, -рівномірність, максимум DDT (таблиці диференціалів), максимум LAT (таблиці лінійних апроксимацій), циклічна структура, наявність фіксованих точок, алгебраїчний імунітет, лінійна збитковість, тощо. Але оскільки для нас в алгоритмі важлива саме швидкість, можна допустити пошук на основі нелінійності.

1. Нелінійність

У статті пропонується обчислювати нелінійність блоків за допомогою перетворення Уолдша-Адамара наступним чином:



У нашому варіанті зміненої реалізації нелінійність блоків обчислюється через максимум таблиці лінійних апроксимацій () наступним чином:

* Будується таблиця лінійних апроксимацій. Кожен елемент у таблиці представляє кількість збігів між лінійним рівнянням, представленим у шістнадцятковій формі як "Вхідна сума", та сумою вихідних бітів, представлених у шістнадцятковій системі у вигляді "Вихідної суми" – 8 (для S-Box’ів розмірності ).
* Знаходиться абсолютний максимум таблиці.
* Обчислюється нелінійність за формулою:



де  - розмірність S-Box’у.

**Алгоритм**

У даному розділі описується запропонований у статті алгоритм генерації вузлів заміни за допомогою методу Particle Swarm Optimization, а також пропонується та описується модифікований алгоритм, який дає значно кращі результати, ніж ті, що надаються в статті.

Опис алгоритму зі статті:

* Ініціалізація популяції

У випадку задачі оптимізації S-box’ів, кожен блок  приймається за частку. Популяція S-box’ів генерується випадковим методом, таким чином, щоб S-Box’и залишалися бієктивними. Генерація повторюється доти, доки популяція розміру  не буде заповнено.

* Обчислення нелінійності

Далі для кожного блоку обчислюється нелінійність. Згідно з цією нелінійністю сортується сама популяція блоків (за спаданням нелінійності).

* Ініціалізація вектору PSO

Вектор швидкості заповнюється нулями і оновлюється на кожній вдалій ітерації. Ініціалізація кожного вектору розташування відбувається з використанням відповідного S-Box’у в популяції. Вектор швидкості оновлюється за формулою (1.1), а вектор розташування оновлюється за формулою (1.2). Вектор найбільш оптимістичних позицій під час виконання ітерації () оновлюється для кожної згенерованої популяції, якщо значення нелінійності нових блоків є кращими за попередні. Найбільш оптимальною з усіх є частинка з найбільшою нелінійністю у популяції.

* Ініціалізація параметрів PSO

Параметри PSO, такі як ,, та  обираються випадковим чином за допомогою вибірки Рені. Протягом оптимізаційної фази роботи алгоритму ці параметри випадково змінюються на кожній ітерації. Ще одним параметром є коефіцієнт інерційності (інерційна вага), який задається за формулою:



де  та  є початковим та кінцевим значенням коефіцієнту відповідно.

* Покращення та регулювання

Вектори швидкості та розташування оновлюються згідно з законами в формулах (1.1) та (1.2) відповідно. Процес такого покращення генерує певні значення, які повторюються або ж від’ємні значення. Щоб цьому зарадити, автори використовують певні методи та процеси обробки, замінюючи повторні значення, значеннями, яких не вистачає для збереження бієктивності. Алгоритм таких покращень у статті не надано.

* Фінальний крок ітерації

Далі для всіх нових згенерованих блоків також обчислюється значення нелінійності, всі S-Box’и, включаючи ті, що вже знаходилися в популяції, знову сортуються за спаданням нелінійності. В популяції залишаються  кращих за нелінійністю S-Box’ів, всі інші блоки відкидаються. Вектори  та  оновлюються як вказано вище.

Псевдокод алгоритму, представленого у статті, показано далі:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вхідні значення:

N ← число блоків в популяції

max\_itr ← максимальна кількість ітерацій

xr ← початкове значення вибірки Рені

c ← параметр вибірки Рені

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Генерація початкової популяції S-box’ів:

xr ← renyi\_map(xr, c, 100)

population ← zeros(2 × N, 256) // 256 для 8 × 8 S-box’ів

for i ← 1 to N

[sboxi, xr] ← gen\_sbox(xr, c)

population[i] ← sboxi

end for

population[1] ← aes\_sbox

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обчислення нелінійності для кожної з часток:

for i ← 1 to N do:

NL[i] ← nonlinearity(population[i])

end for

NL\_sorted ← sort(NL) // сортування у порядку спадання

gBest ← population[1]

pBesti ← population[i]

Vel ← zeros(N, 256)

Встановлення інерційної ваги w

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Початок фази покращення:

While (max\_itr > 0) do:

xr ← renyi\_map(xr, c); c1 ← 2∗xr

xr ← renyi\_map(xr, c); c2 ← 2∗xr

xr ← renyi\_map(xr, c); r1 ← xr

xr ← renyi\_map(xr, c); r2 ← xr

NL ← NL\_sorted

for i ← 1 to N do:

for j = 1 to 256 do:

Vel[i][j] ← ceil(w∗Vel[i][j] + c1∗r1∗(pBest[i][j] - population[i][j]) + c2∗r2∗(gBest[j]-population[i][j]))

if (Vel[i][j] < 0)

Vel[i][j] ← (Vel[i][j] + 256)mod(256)

end if

X[i, j] ← int(population[i][j] + Vel[i][j])mod(256)

temp\_sbox[j] ← X[i, j]

end for

Застосування певного алгоритму регулювання для збереження бієктивності temp\_sbox’у

population[N + i] ← temp\_sbox

end for

for i ← 1 to N do:

NL\_sorted[N + i] ← nonlinearity(population[N + i])

end for

NL\_sorted ← sort(NL\_sorted)

Відкидання зайвих у популяції елементів згідно з нелінійністю

for i ← 1 to N do:

if (NL[i] < NL\_sorted[i])

pBest[i] ← population[i]

end if

end for

Зміна gBest, якщо такий знайдено

max\_itr ← max\_itr - 1

end while

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Алгоритм 1 - PSO зі статті

Модифікований метод майже повністю повторює запропонований алгоритм, різниця полягає в першій ітерації циклу while при формуванні першого блоку в новій популяції, а також при заповненні векторів найбільш оптимальних вузлів. Опис модифікованого алгоритму наведено нижче:

* Ініціалізація популяції

Як і в розглянутому раніше алгоритмі, кожен блок  приймається за частку. Популяція S-box’ів генерується випадковим методом, таким чином, щоб S-Box’и залишалися бієктивними. Генерація повторюється доти, доки популяція розміру  не буде заповнено.

* Обчислення нелінійності

Далі для кожного блоку обчислюється нелінійність. Згідно з цією нелінійністю сортується сама популяція блоків (за спаданням нелінійності).

* Ініціалізація вектору PSO

Вектор швидкості заповнюється нулями і оновлюється на кожній вдалій ітерації. Ініціалізація кожного вектору розташування відбувається з використанням відповідного S-Box’у в популяції. Вектор швидкості оновлюється за формулою (1.1), а вектор розташування оновлюється за формулою (1.2). Відмінністю в нашому алгоритмі є те, що при формуванні нової популяції перший блок нової популяції формується за рахунок взаємодії першого блоку з початкової популяції з самим собою. А далі, якщо після застосування такого способу нелінійність є більшою ніж 98, блок додатково перемішується випадковим чином, щоб покращити інші параметри, такі як лінійна збитковість, дельта рівномірність та алгебраїчний імунітет. Це робиться для того, щоб досягти певного компромісу, щоб не виникало ситуацій коли один параметр є дуже хорошим, а інші поганими. Завдяки такому методу вдається досягти того, щоб всі параметри були достатніми та задовольняли більшості умов. Вектор найбільш оптимістичних позицій під час виконання ітерації () оновлюється для кожної згенерованої популяції, якщо значення нелінійності нових блоків є кращими за попередні. Найбільш оптимальною з усіх є частинка в нашому методі також оновлюється на кожному кроці і прирівнюється до найкращого блоку у векторі найбільш оптимістичних позицій .

* Ініціалізація параметрів PSO

Параметри PSO, такі як ,, та  обираються випадковим чином. Протягом оптимізаційної фази роботи алгоритму ці параметри випадково змінюються на кожній ітерації. Коефіцієнт інерційності також задається за формулою (1.5).

* Покращення та регулювання

Для збереження бієктивності S-Box’ів було розроблено наступний алгоритм. Після формування блоку кожен його елемент перевіряється на співпадіння з іншим елементом в цьому блоці. Якщо при перевірці певного елементу такий елемент вже є в даному блоці, тоді змінна contains стає одиницею, і значення оновлюється шляхом додавання до нього випадкового значення, та взяття його за модулем 256. Так відбувається доти, доки в S-Box’і залишаться тільки унікальні значення від 0 до 255. Псевдокод алгоритму наведено нижче.

for (int i = 0; i < N; ++i)

for (int j = 0; j < size;)

Оновлення значення за формулою (1.1)

Отримання нового значення X для тимчасового S-Box'у, формула (1.2)

int contains;

if (contains == 0)

tempSbox[j] = X;

end if

else

tempSbox[j] = myModulusDec((tempSbox[j] + rand()), 256);

end else;

contains = 0;

for (int k = 0; k < j; ++k)

if (tempSbox[k] == tempSbox[j])

contains = 1;

break;

end if

end for

if (!contains)

j++;

end if

end for

......

end for

Алгоритм 2 – Алгоритм збереження бієктивності

* Фінальний крок ітерації

Далі для всіх нових згенерованих блоків також обчислюється значення нелінійності, всі S-Box’и, включаючи ті, що вже знаходилися в популяції, знову сортуються за спаданням нелінійності. В популяції залишаються  кращих за нелінійністю S-Box’ів, всі інші блоки відкидаються. Вектори  та  оновлюються як вказано вище.

Псевдокод модифікованого алгоритму наведено нижче:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вхідні значення:

N ← число блоків в популяції

max\_itr ← максимальна кількість ітерацій

mode ← вибір режиму алгоритму (0 – алгоритм зі статті, 1 – модифікований)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Генерація початкової популяції S-box’ів:

srand(time(NULL));

int flag = rand()%size;

population ← zeros(2 × N, 256) // 256 для 8 × 8 S-box’ів

for i ← 1 to N

[sboxi, xr] ← gen\_sbox(i+flag)

Генерація відбувається з використанням rand() та перемішування Фішера-Йейтса

population[i] ← sboxi

end for

population[1] ← aes\_sbox

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обчислення нелінійності для кожної з часток:

for i ← 1 to N do:

NL[i] ← nonlinearity(population[i])

end for

NL\_sorted ← sort(NL) // сортування у порядку спадання

gBest ← population[1]

pBesti ← population[i]

Vel ← zeros(N, 256)

Встановлення інерційної ваги w

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Початок фази покращення:

While (max\_itr > 0) do:

Застосування формули (1.5) для розрахунку ваги

c1 ← 2∗rand() [0,1]

c2 ← 2∗rand() [0,1]

r1 ← rand() [0,1]

r2 ← rand() [0,1]

NL ← NL\_sorted

for i ← 1 to N do:

for j = 1 to 256 do:

if (mode == 1)

Vel[i][j] ← ceil(w∗Vel[i][j] + c1∗r1∗(gBest[i][j] - population[i][j]) + c2∗r2∗(gBest[j]-population[i][j]))

end if

if (mode == 0)

Vel[i][j] ← ceil(w∗Vel[i][j] + c1∗r1∗(pBest[i][j] - population[i][j]) + c2∗r2∗(gBest[j]-population[i][j]))

end if

if (Vel[i][j] < 0)

Vel[i][j] ← (Vel[i][j] + 256)mod(256)

end if

X[i, j] ← int(population[i][j] + Vel[i][j])mod(256)

temp\_sbox[j] ← X[i, j]

end for

Застосування наведеного вище алгоритму для збереження бієктивності

if (i == 0)

int LAT = LATMax(tempSbox, size, count);

int NL = raiseToPower(2, count - 1) - LAT;

if (NL > 98)

for (int v = 0; v < 15; ++v)

srand(tempSbox[v] \* (curIter \* v) % 256);

int coeff = rand() % 50;

printf("coeff1 %d", coeff);

int coeff2 = rand() % 256;

printf("coeff2 %d", coeff2);

int temp = tempSbox[coeff];

tempSbox[coeff] = tempSbox[coeff2];

tempSbox[coeff2] = temp;

end for

end if

end if

population[N + i] ← temp\_sbox

end for

for i ← 1 to N do:

NL\_sorted[N + i] ← nonlinearity(population[N + i])

end for

NL\_sorted ← sort(NL\_sorted)

Відкидання зайвих у популяції елементів згідно з нелінійністю

for i ← 1 to N do:

if (NL[i] < NL\_sorted[i])

pBest[i] ← population[i]

end if

end for

Оцінка всіх блоків в популяції на відповідність необхідним параметрам

Зміна gBest, якщо такий знайдено

max\_itr ← max\_itr – 1

mode = 0

++curIter

end while

Запис кінцевої популяції блоків у файл

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Алгоритм 3 – Модифікований PSO метод

Така модифікація методу дозволяє за порівняно короткий час формувати блоки з нелінійністю 104, лінійною збитковістю 0 та алгебраїчним імунітетом 3.

Розглянемо обидва алгоритми на зрозумілому прикладі, щоб подивитися як вони працюють.

Візьмемо теоретичний S-Box (3,1,0,2) з найкращою можливою теоретичною нелінійністю, який буде взятий за основу, як AES-Sbox в алгоритмі. Наприклад кількість блоків в популяції  - 4. Тому на початку виконання алгоритму буде згенеровано ще 3 випадкові блоки. Цими блоками буде заповнено масив популяції. Теоретична нелінійність буде вказуватися у відсотках від найкращої можливої нелінійності в популяції, тобто 100% - найкраща можлива нелінійність, і чим менше цей відсоток, тим гіршою є нелінійність блоку.

Маємо:

|  |  |
| --- | --- |
| S-Box | Нелінійність |
| (3,1,0,2) | 100% |
| (2,1,3,0) | 83% |
| (1,2,3,0) | 85% |
| (2,3,1,0) | 81% |

Масив 

Після генерації масив буде відсортовано у порядку спадання нелінійності. Отримуємо:

|  |  |
| --- | --- |
| S-Box | Нелінійність |
| (3,1,0,2) | 100% |
| (1,2,3,0) | 85% |
| (2,3,1,0) | 83% |
| (2,1,3,0) | 81% |

Масив 

Після цього заповнюються вектори  та . До вектору  додається перший блок з масиву, тобто блок з найкращою нелінійністю, а до  всі інші блоки. Після цього маємо три масиви:

|  |  |
| --- | --- |
| S-Box | Нелінійність |
| (3,1,0,2) | 100% |

Масив 

|  |  |
| --- | --- |
| S-Box | Нелінійність |
| (1,2,3,0) | 85% |
| (2,3,1,0) | 83% |
| (2,1,3,0) | 81% |

Масив 

|  |  |
| --- | --- |
| S-Box | Нелінійність |
| (3,1,0,2) | 100% |
| (1,2,3,0) | 85% |
| (2,3,1,0) | 83% |
| (2,1,3,0) | 81% |

Масив 

Тепер відбувається ініціалізація інерційної ваги та починається фаза оптимізації. На цьому кроці спочатку генеруються 4 випадкові коефіцієнти, потім за формулою (1.5) змінюємо інерційну вагу та починаємо оновлювати значення вектору швидкості за формулою (1.1).





…



Після формування кожного елементу так, як показано вище, перевіряється, чи не є елемент від’ємним. Якщо елемент все-таки менший за 0, від береться за модулем розмірності блоку, тобто, для блоку , для блоку  і так далі. В нашому випадку алгоритм працює із розмірністю S-Box’ів , а отже всі від’ємні елементи будуть братися за модулем 256.

У прикладі наведено блоки розмірності , тому елемент буде взятий за модулем 4 і отримаємо .

Упускаючи значення коефіцієнтів, припустимо, що ми отримали наступний вектор швидкості:

|  |
| --- |
| Vel |
| (2,1,0,3) |
| (3,2,1,2) |
| (2,3,1,0) |
| (0,1,3,2) |

Масив 

Далі відбувається формування нових блоків, шляхом додавання до значень початкової популяції значень вектору швидкості за формулою (1.2). На цьому кроці також відбувається перевірка того, що значення входять до потрібного діапазону. Якщо значення більше або менше необхідного, воно береться за модулем розмірності блоку.

Також відбувається перевірка на бієктивність і виправлення небієктивності, якщо така є. Процедура виконання перевірки у статті не надавалася.

Після даної процедури ми отримуємо нову популяцію розміру .

Наприклад до старої шляхом модернізації додалося ще  теоретичних блоків з теоретичною нелінійністю і маємо:

|  |  |
| --- | --- |
| S-Box | Нелінійність |
| (3,1,0,2) | 100% |
| (1,2,3,0) | 85% |
| (2,3,1,0) | 83% |
| (2,1,3,0) | 81% |
| (3,2,1,0) | 88% |
| (3,2,0,1) | 85% |
| (2,3,0,1) | 88% |
| (0,1,2,3) | 81% |

Масив 

Далі знову відбувається процедура сортування за нелінійністю популяції.

|  |  |
| --- | --- |
| S-Box | Нелінійність |
| (3,1,0,2) | 100% |
| (3,2,1,0) | 88% |
| (2,3,0,1) | 88% |
| (1,2,3,0) | 85% |
| (3,2,0,1) | 85% |
| (2,3,1,0) | 83% |
| (2,1,3,0) | 81% |
| (0,1,2,3) | 81% |

Масив 

Після сортування останні  блоків відкидаються і масив знову має розмір перед початком наступного кроку оптимізації.

|  |  |
| --- | --- |
| S-Box | Нелінійність |
| (3,1,0,2) | 100% |
| (3,2,1,0) | 88% |
| (3,2,1,0) | 88% |
| (1,2,3,0) | 85% |

Масив 

Далі всі процедури повторюються, заповнюються вектори  та . До вектору  додається перший блок з масиву, а до  всі інші блоки. Це відбувається доти, доки не закінчиться задана кількість ітерацій, або поки не буде знайдено необхідний блок.

Наша версія алгоритму пропонує трохи інший підхід до пошуку блоків. Він полягає в тому, щоб за допомогою часток погіршувати завідома гарний блок для досягнення компромісу між всіма параметрами захищеності. Завдяки такому методу вдається досягти того, щоб всі параметри були достатніми та задовольняли більшості умов.

Отже, модифікований алгоритм працює наступним чином:

Спочатку все відбувається так само, як і у випадку алгоритму, розглянутого раніше: на початку виконання алгоритму у нас є завідома гарний блок, з високою нелінійністю, генеруються ще 3 випадкові блоки. Цими блоками буде заповнено масив популяції. Масив сортується у порядку спадання нелінійності блоків. Заповнюються вектори  та .

Відмінністю методу є те, що на першому кроці алгоритму відбувається перемішування першого блоку в популяції, щоб не сильно погіршити одні його показники та покращити інші і на його основі вже далі погіршувати одні показники та покращувати інші.

Припустимо, як і в минулому прикладі, у нас на початку першого кроку генерується популяція:

|  |  |
| --- | --- |
| S-Box | Нелінійність |
| (3,1,0,2) | 100% |
| (1,2,3,0) | 85% |
| (2,3,1,0) | 83% |
| (2,1,3,0) | 81% |

Масив 

Після цього невелика кількість значень з найкращого блоку перемішуються, щоб погіршити нелінійність не дуже значним чином. Отримуємо:

|  |  |
| --- | --- |
| S-Box | Нелінійність |
| (3,1,0,2) | 100% |
| (3,1,2,0) | 95% |
| (2,3,1,0) | 83% |
| (2,1,3,0) | 81% |

Масив 

Отриманий таким чином блок має гарну нелінійність, але всі інші параметри, такі як алгебраїчний імунітет, дельта рівномірність та лінійна збитковість є поганими, тому далі продовжується оптимізація популяції, як в алгоритмі, який описано вище. Всі наступні ітерації ніяк не відрізняються від алгоритму, запропонованого вище.

Оптимізація відбувається доти, доки не закінчиться задана кількість ітерацій, або поки не буде знайдено необхідний блок. В нашому випадку алгоритм здатний знаходити S-Box’и з нелінійністю 104 та іншими показниками, які задовольняють умовам захищеності.

Тепер розглянувши спрощену модель обох алгоритмів, пропонуємо ознайомитися з реальними прикладами виконання програми для блоків заміни розмірності , тобто довжини 256. Для прикладу візьмемо розмір популяції рівним 4 – тобто . Спочатку розглянемо алгоритм зі статті під час реального виконання.

При  на початку виконання алгоритму буде згенеровано ще 3 випадкові блоки. Цими блоками буде заповнено масив популяції. Маємо:

|  |  |
| --- | --- |
| S-Box | NL |
| (99, 124, 119, 123, 242, 107, 111, 197, 48, 1, 103, 43, 254, 215, 171, 118,  202, 130, 201, 125, 250, 89, 71, 240, 173, 212, 162, 175, 156, 164, 114, 192,  183, 253, 147, 38, 54, 63, 247, 204, 52, 165, 229, 241, 113, 216, 49, 21,  4, 199, 35, 195, 24, 150, 5, 154, 7, 18, 128, 226, 235, 39, 178, 117,  9, 131, 44, 26, 27, 110, 90, 160, 82, 59, 214, 179, 41, 227, 47, 132,  83, 209, 0, 237, 32, 252, 177, 91, 106, 203, 190, 57, 74, 76, 88, 207,  208, 239, 170, 251, 67, 77, 51, 133, 69, 249, 2, 127, 80, 60, 159, 168,  81, 163, 64, 143, 146, 157, 56, 245, 188, 182, 218, 33, 16, 255, 243, 210,  205, 12, 19, 236, 95, 151, 68, 23, 196, 167, 126, 61, 100, 93, 25, 115,  96, 129, 79, 220, 34, 42, 144, 136, 70, 238, 184, 20, 222, 94, 11, 219,  224, 50, 58, 10, 73, 6, 36, 92, 194, 211, 172, 98, 145, 149, 228, 121,  231, 200, 55, 109, 141, 213, 78, 169, 108, 86, 244, 234, 101, 122, 174, 8,  186, 120, 37, 46, 28, 166, 180, 198, 232, 221, 116, 31, 75, 189, 139, 138,  112, 62, 181, 102, 72, 3, 246, 14, 97, 53, 87, 185, 134, 193, 29, 158,  225, 248, 152, 17, 105, 217, 142, 148, 155, 30, 135, 233, 206, 85, 40, 223,  140, 161, 137, 13, 191, 230, 66, 104, 65, 153, 45, 15, 176, 84, 187, 22) | 112 |
| (170, 185, 249, 145, 2, 13, 179, 208, 28, 196, 168, 80, 124, 112, 113, 243,  74, 238, 160, 65, 30, 251, 7, 33, 234, 161, 32, 61, 19, 206, 230, 165,  29, 131, 186, 236, 190, 0, 72, 106, 95, 96, 15, 193, 212, 98, 52, 53,  150, 26, 69, 36, 217, 205, 192, 246, 233, 56, 101, 232, 201, 58, 252, 1,  127, 121, 227, 68, 180, 31, 203, 91, 102, 147, 173, 167, 25, 250, 211, 221,  187, 47, 59, 88, 35, 129, 139, 235, 222, 132, 142, 105, 240, 120, 143, 34,  75, 245, 55, 116, 119, 93, 157, 27, 66, 8, 39, 18, 77, 89, 228, 4,  148, 51, 200, 14, 87, 130, 63, 141, 184, 151, 226, 152, 183, 43, 215, 163,  37, 254, 70, 46, 79, 117, 153, 242, 149, 138, 73, 100, 82, 45, 210, 97,  44, 162, 213, 40, 146, 241, 78, 83, 99, 41, 253, 92, 16, 209, 110, 64,  133, 189, 175, 174, 255, 54, 107, 118, 103, 22, 140, 123, 158, 42, 191, 49,  195, 84, 6, 9, 229, 21, 136, 60, 220, 94, 115, 48, 81, 90, 126, 111,  108, 216, 50, 197, 17, 198, 122, 20, 202, 207, 85, 231, 134, 24, 218, 128,  11, 166, 194, 38, 164, 199, 12, 177, 224, 3, 10, 172, 223, 156, 247, 239,  23, 181, 104, 182, 219, 188, 248, 214, 71, 155, 204, 144, 67, 57, 159, 237,  178, 5, 225, 76, 109, 169, 137, 135, 62, 125, 176, 86, 244, 154, 114, 171) | 92 |
| (72, 67, 68, 107, 189, 235, 163, 243, 96, 102, 93, 224, 229, 101, 215, 152,  204, 222, 0, 131, 126, 244, 46, 193, 217, 64, 43, 187, 181, 228, 210, 221,  70, 6, 122, 164, 156, 39, 245, 172, 239, 237, 160, 192, 91, 167, 135, 9,  97, 207, 158, 50, 177, 203, 66, 145, 86, 194, 42, 254, 179, 240, 90, 38,  144, 208, 130, 117, 166, 139, 120, 59, 185, 53, 44, 115, 75, 150, 73, 143,  176, 116, 137, 227, 255, 129, 89, 47, 238, 3, 151, 242, 188, 209, 79, 231,  8, 214, 31, 200, 52, 87, 63, 230, 219, 11, 20, 213, 35, 170, 161, 108,  250, 132, 123, 212, 121, 140, 169, 36, 78, 196, 146, 25, 118, 232, 5, 28,  211, 220, 65, 112, 128, 45, 141, 119, 19, 32, 162, 92, 216, 69, 24, 218,  186, 54, 2, 76, 110, 241, 29, 56, 171, 14, 183, 147, 233, 182, 201, 103,  149, 81, 58, 246, 226, 82, 40, 17, 175, 159, 155, 114, 248, 251, 1, 125,  142, 74, 136, 197, 100, 133, 157, 109, 199, 94, 180, 198, 191, 80, 178, 16,  34, 234, 62, 12, 60, 48, 23, 27, 71, 85, 148, 124, 206, 168, 88, 225,  30, 22, 61, 165, 205, 154, 173, 13, 249, 247, 83, 98, 153, 104, 57, 49,  84, 4, 10, 77, 21, 252, 51, 33, 95, 202, 99, 37, 184, 18, 190, 195,  7, 138, 127, 134, 253, 236, 111, 174, 223, 55, 106, 113, 105, 26, 41, 15) | 94 |
| (242, 201, 159, 34, 75, 78, 207, 185, 225, 116, 94, 1, 140, 86, 232, 93,  209, 131, 165, 69, 177, 246, 194, 99, 171, 228, 70, 154, 199, 189, 80, 60,  250, 182, 144, 151, 241, 87, 100, 4, 183, 23, 252, 253, 25, 3, 179, 126,  68, 134, 195, 90, 218, 27, 57, 190, 222, 149, 143, 6, 184, 192, 198, 238,  55, 239, 114, 81, 254, 89, 36, 14, 37, 33, 56, 135, 213, 148, 203, 217,  124, 2, 88, 216, 113, 233, 255, 101, 146, 43, 142, 103, 188, 73, 175, 191,  123, 46, 84, 173, 247, 202, 214, 102, 138, 51, 166, 109, 119, 64, 145, 115,  244, 107, 163, 150, 193, 18, 221, 0, 136, 160, 17, 155, 62, 40, 223, 117,  49, 83, 229, 13, 118, 147, 54, 132, 7, 245, 226, 227, 63, 12, 161, 208,  9, 24, 121, 156, 48, 237, 157, 112, 235, 204, 15, 104, 71, 176, 67, 215,  219, 95, 120, 168, 79, 110, 251, 169, 22, 44, 122, 231, 197, 58, 76, 230,  139, 137, 97, 59, 128, 74, 186, 248, 11, 162, 66, 111, 133, 52, 19, 29,  42, 236, 98, 39, 10, 172, 181, 240, 45, 105, 206, 61, 16, 65, 85, 130,  205, 92, 47, 125, 167, 153, 180, 50, 243, 174, 127, 41, 234, 249, 152, 106,  28, 96, 196, 108, 38, 129, 5, 170, 26, 77, 178, 82, 31, 32, 20, 210,  21, 141, 91, 8, 158, 187, 72, 164, 211, 53, 35, 220, 30, 200, 224, 212) | 94 |

Масив 

Після генерації масив буде відсортовано у порядку спадання нелінійності. Отримуємо:

|  |  |
| --- | --- |
| S-Box | NL |
| (99, 124, 119, 123, 242, 107, 111, 197, 48, 1, 103, 43, 254, 215, 171, 118,  202, 130, 201, 125, 250, 89, 71, 240, 173, 212, 162, 175, 156, 164, 114, 192,  183, 253, 147, 38, 54, 63, 247, 204, 52, 165, 229, 241, 113, 216, 49, 21,  4, 199, 35, 195, 24, 150, 5, 154, 7, 18, 128, 226, 235, 39, 178, 117,  9, 131, 44, 26, 27, 110, 90, 160, 82, 59, 214, 179, 41, 227, 47, 132,  83, 209, 0, 237, 32, 252, 177, 91, 106, 203, 190, 57, 74, 76, 88, 207,  208, 239, 170, 251, 67, 77, 51, 133, 69, 249, 2, 127, 80, 60, 159, 168,  81, 163, 64, 143, 146, 157, 56, 245, 188, 182, 218, 33, 16, 255, 243, 210,  205, 12, 19, 236, 95, 151, 68, 23, 196, 167, 126, 61, 100, 93, 25, 115,  96, 129, 79, 220, 34, 42, 144, 136, 70, 238, 184, 20, 222, 94, 11, 219,  224, 50, 58, 10, 73, 6, 36, 92, 194, 211, 172, 98, 145, 149, 228, 121,  231, 200, 55, 109, 141, 213, 78, 169, 108, 86, 244, 234, 101, 122, 174, 8,  186, 120, 37, 46, 28, 166, 180, 198, 232, 221, 116, 31, 75, 189, 139, 138,  112, 62, 181, 102, 72, 3, 246, 14, 97, 53, 87, 185, 134, 193, 29, 158,  225, 248, 152, 17, 105, 217, 142, 148, 155, 30, 135, 233, 206, 85, 40, 223,  140, 161, 137, 13, 191, 230, 66, 104, 65, 153, 45, 15, 176, 84, 187, 22) | 112 |
| (72, 67, 68, 107, 189, 235, 163, 243, 96, 102, 93, 224, 229, 101, 215, 152,  204, 222, 0, 131, 126, 244, 46, 193, 217, 64, 43, 187, 181, 228, 210, 221,  70, 6, 122, 164, 156, 39, 245, 172, 239, 237, 160, 192, 91, 167, 135, 9,  97, 207, 158, 50, 177, 203, 66, 145, 86, 194, 42, 254, 179, 240, 90, 38,  144, 208, 130, 117, 166, 139, 120, 59, 185, 53, 44, 115, 75, 150, 73, 143,  176, 116, 137, 227, 255, 129, 89, 47, 238, 3, 151, 242, 188, 209, 79, 231,  8, 214, 31, 200, 52, 87, 63, 230, 219, 11, 20, 213, 35, 170, 161, 108,  250, 132, 123, 212, 121, 140, 169, 36, 78, 196, 146, 25, 118, 232, 5, 28,  211, 220, 65, 112, 128, 45, 141, 119, 19, 32, 162, 92, 216, 69, 24, 218,  186, 54, 2, 76, 110, 241, 29, 56, 171, 14, 183, 147, 233, 182, 201, 103,  149, 81, 58, 246, 226, 82, 40, 17, 175, 159, 155, 114, 248, 251, 1, 125,  142, 74, 136, 197, 100, 133, 157, 109, 199, 94, 180, 198, 191, 80, 178, 16,  34, 234, 62, 12, 60, 48, 23, 27, 71, 85, 148, 124, 206, 168, 88, 225,  30, 22, 61, 165, 205, 154, 173, 13, 249, 247, 83, 98, 153, 104, 57, 49,  84, 4, 10, 77, 21, 252, 51, 33, 95, 202, 99, 37, 184, 18, 190, 195,  7, 138, 127, 134, 253, 236, 111, 174, 223, 55, 106, 113, 105, 26, 41, 15) | 94 |
| (242, 201, 159, 34, 75, 78, 207, 185, 225, 116, 94, 1, 140, 86, 232, 93,  209, 131, 165, 69, 177, 246, 194, 99, 171, 228, 70, 154, 199, 189, 80, 60,  250, 182, 144, 151, 241, 87, 100, 4, 183, 23, 252, 253, 25, 3, 179, 126,  68, 134, 195, 90, 218, 27, 57, 190, 222, 149, 143, 6, 184, 192, 198, 238,  55, 239, 114, 81, 254, 89, 36, 14, 37, 33, 56, 135, 213, 148, 203, 217,  124, 2, 88, 216, 113, 233, 255, 101, 146, 43, 142, 103, 188, 73, 175, 191,  123, 46, 84, 173, 247, 202, 214, 102, 138, 51, 166, 109, 119, 64, 145, 115,  244, 107, 163, 150, 193, 18, 221, 0, 136, 160, 17, 155, 62, 40, 223, 117,  49, 83, 229, 13, 118, 147, 54, 132, 7, 245, 226, 227, 63, 12, 161, 208,  9, 24, 121, 156, 48, 237, 157, 112, 235, 204, 15, 104, 71, 176, 67, 215,  219, 95, 120, 168, 79, 110, 251, 169, 22, 44, 122, 231, 197, 58, 76, 230,  139, 137, 97, 59, 128, 74, 186, 248, 11, 162, 66, 111, 133, 52, 19, 29,  42, 236, 98, 39, 10, 172, 181, 240, 45, 105, 206, 61, 16, 65, 85, 130,  205, 92, 47, 125, 167, 153, 180, 50, 243, 174, 127, 41, 234, 249, 152, 106,  28, 96, 196, 108, 38, 129, 5, 170, 26, 77, 178, 82, 31, 32, 20, 210,  21, 141, 91, 8, 158, 187, 72, 164, 211, 53, 35, 220, 30, 200, 224, 212) | 94 |
| (170, 185, 249, 145, 2, 13, 179, 208, 28, 196, 168, 80, 124, 112, 113, 243,  74, 238, 160, 65, 30, 251, 7, 33, 234, 161, 32, 61, 19, 206, 230, 165,  29, 131, 186, 236, 190, 0, 72, 106, 95, 96, 15, 193, 212, 98, 52, 53,  150, 26, 69, 36, 217, 205, 192, 246, 233, 56, 101, 232, 201, 58, 252, 1,  127, 121, 227, 68, 180, 31, 203, 91, 102, 147, 173, 167, 25, 250, 211, 221,  187, 47, 59, 88, 35, 129, 139, 235, 222, 132, 142, 105, 240, 120, 143, 34,  75, 245, 55, 116, 119, 93, 157, 27, 66, 8, 39, 18, 77, 89, 228, 4,  148, 51, 200, 14, 87, 130, 63, 141, 184, 151, 226, 152, 183, 43, 215, 163,  37, 254, 70, 46, 79, 117, 153, 242, 149, 138, 73, 100, 82, 45, 210, 97,  44, 162, 213, 40, 146, 241, 78, 83, 99, 41, 253, 92, 16, 209, 110, 64,  133, 189, 175, 174, 255, 54, 107, 118, 103, 22, 140, 123, 158, 42, 191, 49,  195, 84, 6, 9, 229, 21, 136, 60, 220, 94, 115, 48, 81, 90, 126, 111,  108, 216, 50, 197, 17, 198, 122, 20, 202, 207, 85, 231, 134, 24, 218, 128,  11, 166, 194, 38, 164, 199, 12, 177, 224, 3, 10, 172, 223, 156, 247, 239,  23, 181, 104, 182, 219, 188, 248, 214, 71, 155, 204, 144, 67, 57, 159, 237,  178, 5, 225, 76, 109, 169, 137, 135, 62, 125, 176, 86, 244, 154, 114, 171) | 92 |

Масив 

Після цього заповнюються вектори  та . До вектору  додається перший блок з масиву, тобто блок з найкращою нелінійністю, а до  всі інші блоки. Після цього маємо три масиви:

|  |  |
| --- | --- |
| S-Box | NL |
| (99, 124, 119, 123, 242, 107, 111, 197, 48, 1, 103, 43, 254, 215, 171, 118,  202, 130, 201, 125, 250, 89, 71, 240, 173, 212, 162, 175, 156, 164, 114, 192,  183, 253, 147, 38, 54, 63, 247, 204, 52, 165, 229, 241, 113, 216, 49, 21,  4, 199, 35, 195, 24, 150, 5, 154, 7, 18, 128, 226, 235, 39, 178, 117,  9, 131, 44, 26, 27, 110, 90, 160, 82, 59, 214, 179, 41, 227, 47, 132,  83, 209, 0, 237, 32, 252, 177, 91, 106, 203, 190, 57, 74, 76, 88, 207,  208, 239, 170, 251, 67, 77, 51, 133, 69, 249, 2, 127, 80, 60, 159, 168,  81, 163, 64, 143, 146, 157, 56, 245, 188, 182, 218, 33, 16, 255, 243, 210,  205, 12, 19, 236, 95, 151, 68, 23, 196, 167, 126, 61, 100, 93, 25, 115,  96, 129, 79, 220, 34, 42, 144, 136, 70, 238, 184, 20, 222, 94, 11, 219,  224, 50, 58, 10, 73, 6, 36, 92, 194, 211, 172, 98, 145, 149, 228, 121,  231, 200, 55, 109, 141, 213, 78, 169, 108, 86, 244, 234, 101, 122, 174, 8,  186, 120, 37, 46, 28, 166, 180, 198, 232, 221, 116, 31, 75, 189, 139, 138,  112, 62, 181, 102, 72, 3, 246, 14, 97, 53, 87, 185, 134, 193, 29, 158,  225, 248, 152, 17, 105, 217, 142, 148, 155, 30, 135, 233, 206, 85, 40, 223,  140, 161, 137, 13, 191, 230, 66, 104, 65, 153, 45, 15, 176, 84, 187, 22) | 112 |

Масив 

|  |  |
| --- | --- |
| S-Box | NL |
| (72, 67, 68, 107, 189, 235, 163, 243, 96, 102, 93, 224, 229, 101, 215, 152,  204, 222, 0, 131, 126, 244, 46, 193, 217, 64, 43, 187, 181, 228, 210, 221,  70, 6, 122, 164, 156, 39, 245, 172, 239, 237, 160, 192, 91, 167, 135, 9,  97, 207, 158, 50, 177, 203, 66, 145, 86, 194, 42, 254, 179, 240, 90, 38,  144, 208, 130, 117, 166, 139, 120, 59, 185, 53, 44, 115, 75, 150, 73, 143,  176, 116, 137, 227, 255, 129, 89, 47, 238, 3, 151, 242, 188, 209, 79, 231,  8, 214, 31, 200, 52, 87, 63, 230, 219, 11, 20, 213, 35, 170, 161, 108,  250, 132, 123, 212, 121, 140, 169, 36, 78, 196, 146, 25, 118, 232, 5, 28,  211, 220, 65, 112, 128, 45, 141, 119, 19, 32, 162, 92, 216, 69, 24, 218,  186, 54, 2, 76, 110, 241, 29, 56, 171, 14, 183, 147, 233, 182, 201, 103,  149, 81, 58, 246, 226, 82, 40, 17, 175, 159, 155, 114, 248, 251, 1, 125,  142, 74, 136, 197, 100, 133, 157, 109, 199, 94, 180, 198, 191, 80, 178, 16,  34, 234, 62, 12, 60, 48, 23, 27, 71, 85, 148, 124, 206, 168, 88, 225,  30, 22, 61, 165, 205, 154, 173, 13, 249, 247, 83, 98, 153, 104, 57, 49,  84, 4, 10, 77, 21, 252, 51, 33, 95, 202, 99, 37, 184, 18, 190, 195,  7, 138, 127, 134, 253, 236, 111, 174, 223, 55, 106, 113, 105, 26, 41, 15) | 94 |
| (242, 201, 159, 34, 75, 78, 207, 185, 225, 116, 94, 1, 140, 86, 232, 93,  209, 131, 165, 69, 177, 246, 194, 99, 171, 228, 70, 154, 199, 189, 80, 60,  250, 182, 144, 151, 241, 87, 100, 4, 183, 23, 252, 253, 25, 3, 179, 126,  68, 134, 195, 90, 218, 27, 57, 190, 222, 149, 143, 6, 184, 192, 198, 238,  55, 239, 114, 81, 254, 89, 36, 14, 37, 33, 56, 135, 213, 148, 203, 217,  124, 2, 88, 216, 113, 233, 255, 101, 146, 43, 142, 103, 188, 73, 175, 191,  123, 46, 84, 173, 247, 202, 214, 102, 138, 51, 166, 109, 119, 64, 145, 115,  244, 107, 163, 150, 193, 18, 221, 0, 136, 160, 17, 155, 62, 40, 223, 117,  49, 83, 229, 13, 118, 147, 54, 132, 7, 245, 226, 227, 63, 12, 161, 208,  9, 24, 121, 156, 48, 237, 157, 112, 235, 204, 15, 104, 71, 176, 67, 215,  219, 95, 120, 168, 79, 110, 251, 169, 22, 44, 122, 231, 197, 58, 76, 230,  139, 137, 97, 59, 128, 74, 186, 248, 11, 162, 66, 111, 133, 52, 19, 29,  42, 236, 98, 39, 10, 172, 181, 240, 45, 105, 206, 61, 16, 65, 85, 130,  205, 92, 47, 125, 167, 153, 180, 50, 243, 174, 127, 41, 234, 249, 152, 106,  28, 96, 196, 108, 38, 129, 5, 170, 26, 77, 178, 82, 31, 32, 20, 210,  21, 141, 91, 8, 158, 187, 72, 164, 211, 53, 35, 220, 30, 200, 224, 212) | 94 |
| (170, 185, 249, 145, 2, 13, 179, 208, 28, 196, 168, 80, 124, 112, 113, 243,  74, 238, 160, 65, 30, 251, 7, 33, 234, 161, 32, 61, 19, 206, 230, 165,  29, 131, 186, 236, 190, 0, 72, 106, 95, 96, 15, 193, 212, 98, 52, 53,  150, 26, 69, 36, 217, 205, 192, 246, 233, 56, 101, 232, 201, 58, 252, 1,  127, 121, 227, 68, 180, 31, 203, 91, 102, 147, 173, 167, 25, 250, 211, 221,  187, 47, 59, 88, 35, 129, 139, 235, 222, 132, 142, 105, 240, 120, 143, 34,  75, 245, 55, 116, 119, 93, 157, 27, 66, 8, 39, 18, 77, 89, 228, 4,  148, 51, 200, 14, 87, 130, 63, 141, 184, 151, 226, 152, 183, 43, 215, 163,  37, 254, 70, 46, 79, 117, 153, 242, 149, 138, 73, 100, 82, 45, 210, 97,  44, 162, 213, 40, 146, 241, 78, 83, 99, 41, 253, 92, 16, 209, 110, 64,  133, 189, 175, 174, 255, 54, 107, 118, 103, 22, 140, 123, 158, 42, 191, 49,  195, 84, 6, 9, 229, 21, 136, 60, 220, 94, 115, 48, 81, 90, 126, 111,  108, 216, 50, 197, 17, 198, 122, 20, 202, 207, 85, 231, 134, 24, 218, 128,  11, 166, 194, 38, 164, 199, 12, 177, 224, 3, 10, 172, 223, 156, 247, 239,  23, 181, 104, 182, 219, 188, 248, 214, 71, 155, 204, 144, 67, 57, 159, 237,  178, 5, 225, 76, 109, 169, 137, 135, 62, 125, 176, 86, 244, 154, 114, 171) | 92 |

Масив 

Тепер відбувається ініціалізація інерційної ваги та починається фаза оптимізації. На цьому кроці спочатку генеруються 4 випадкові коефіцієнти, потім за формулою (1.5) змінюємо інерційну вагу та починаємо оновлювати значення вектору швидкості за формулою (1.1).

Випадкові значення коефіцієнтів: c1 = 1,5, c2 = 0,68, r1 = 0,07, r2 = 0,83. Елементи вектору, як вже згадувалося раніше, формуються за законом з формули (1.1). Після формування кожного елементу перевіряється, чи не є елемент від’ємним і, якщо елемент менше 0, то він береться за модулем 256. Отримали наступний вектор швидкості:

|  |
| --- |
| Vel |
| (254, 251, 251, 255, 251, 14, 6, 5, 6, 11, 255, 20, 254, 245, 5, 254, 1, 10, 235, 1, 159, 17, 254, 252, 5, 241, 158, 2, 3, 7, 11, 4, 245, 231, 254, 14, 11, 254, 0, 253, 20, 8, 249, 156, 254, 251, 10, 255, 10, 1, 13, 160, 17, 6, 253, 0, 9, 19, 158, 3, 251, 247, 157, 248, 238, 252, 252, 10, 15, 160, 4, 26, 11, 0, 160, 31, 4, 157, 3, 2, 252, 27, 15, 154, 24, 22, 247, 155, 14, 235, 252, 20, 12, 14, 0, 3, 235, 154, 159, 251, 255, 2, 2, 11, 16, 232, 0, 10, 252, 173, 1, 32, 18, 36, 7, 8, 154, 162, 12, 235, 24, 2, 31, 0, 11, 254, 3, 162, 1, 22, 162, 159, 160, 158, 214, 251, 161, 242, 4, 160, 13, 189, 0, 171, 169, 156, 208, 241, 253, 186, 244, 248, 11, 173, 0, 176, 204, 10, 20, 244, 249, 36, 0, 25, 9, 224, 0, 156, 255, 205, 39, 2, 28, 170, 224, 0, 209, 159, 9, 169, 35, 211, 252, 250, 17, 0, 250, 253, 207, 155, 1, 1, 242, 252, 3, 203, 4, 213, 160, 239, 193, 232, 223, 252, 14, 237, 220, 197, 211, 179, 239, 206, 251, 4, 207, 0, 178, 21, 0, 208, 2, 27, 217, 245, 159, 19, 242, 184, 248, 0, 247, 159, 250, 20, 192, 236, 12, 249, 250, 230, 158, 6, 37, 13, 208, 0, 221, 223, 18, 149, 32, 4, 238, 163, 16, 0) |
| (20, 18, 13, 250, 248, 232, 2, 248, 11, 252, 1, 222, 249, 163, 0, 248, 1, 241, 30, 250, 13, 248, 18, 249, 249, 26, 10, 252, 1, 249, 237, 238, 26, 34, 4, 248, 3, 7, 241, 241, 240, 164, 14, 10, 251, 242, 0, 13, 248, 248, 253, 13, 252, 162, 252, 6, 252, 241, 16, 229, 255, 240, 249, 246, 239, 40, 250, 247, 2, 250, 246, 2, 235, 255, 12, 6, 13, 254, 250, 252, 246, 156, 159, 0, 165, 248, 23, 168, 160, 17, 255, 231, 31, 234, 252, 251, 24, 240, 14, 1, 22, 12, 16, 222, 169, 184, 15, 240, 12, 160, 255, 5, 246, 0, 0, 158, 252, 245, 154, 31, 13, 155, 247, 177, 158, 238, 22, 21, 239, 0, 15, 253, 36, 23, 159, 252, 10, 243, 211, 13, 224, 252, 250, 157, 233, 255, 184, 17, 158, 244, 21, 29, 1, 34, 160, 22, 15, 222, 231, 19, 199, 0, 165, 6, 163, 231, 248, 0, 0, 170, 201, 251, 245, 26, 247, 253, 246, 23, 204, 237, 253, 188, 132, 172, 108, 165, 8, 202, 245, 158, 240, 0, 9, 180, 158, 60, 30, 247, 11, 76, 217, 202, 254, 194, 128, 27, 158, 3, 7, 169, 215, 30, 233, 242, 163, 38, 209, 237, 212, 0, 223, 186, 9, 193, 3, 254, 20, 0, 32, 242, 0, 15, 202, 139, 247, 247, 0, 230, 196, 184, 218, 192, 200, 195, 15, 251, 168, 178, 209, 210, 7, 3, 155, 247, 16, 184) |
| (235, 216, 0, 116, 6, 34, 0, 136, 69, 161, 218, 11, 4, 11, 0, 198, 238, 6, 187, 1, 245, 241, 179, 2, 2, 180, 138, 148, 3, 187, 194, 26, 204, 0, 5, 38, 240, 246, 6, 23, 161, 17, 230, 249, 25, 23, 235, 243, 5, 249, 162, 1, 158, 26, 12, 4, 245, 239, 251, 37, 254, 233, 5, 224, 5, 12, 8, 155, 235, 252, 21, 17, 10, 250, 22, 6, 227, 16, 248, 252, 5, 17, 248, 244, 213, 247, 240, 14, 254, 19, 3, 154, 255, 253, 248, 241, 0, 243, 3, 255, 232, 14, 241, 156, 158, 8, 233, 157, 217, 3, 10, 248, 11, 254, 255, 242, 243, 20, 164, 30, 252, 255, 34, 249, 10, 14, 255, 11, 253, 251, 227, 249, 156, 38, 12, 163, 27, 241, 235, 163, 254, 9, 200, 161, 9, 21, 167, 191, 10, 212, 158, 255, 3, 218, 210, 250, 4, 154, 38, 213, 157, 166, 208, 248, 16, 0, 0, 247, 0, 247, 208, 237, 249, 30, 10, 245, 185, 39, 22, 155, 12, 0, 141, 253, 1, 245, 208, 0, 213, 67, 186, 255, 177, 180, 248, 205, 0, 33, 229, 190, 245, 17, 161, 188, 16, 209, 19, 219, 132, 163, 24, 201, 176, 252, 245, 172, 209, 164, 187, 23, 249, 243, 207, 247, 230, 18, 20, 161, 27, 6, 20, 1, 185, 252, 174, 229, 15, 212, 64, 4, 9, 190, 231, 215, 217, 0, 214, 205, 250, 0, 6, 2, 26, 219, 156, 207) |
| (229, 45, 0, 82, 17, 146, 0, 111, 244, 222, 35, 246, 250, 251, 0, 113, 160, 222, 225, 252, 10, 176, 209, 9, 168, 174, 0, 1, 247, 4, 225, 196, 218, 177, 221, 220, 165, 4, 3, 251, 244, 251, 12, 239, 228, 253, 251, 249, 232, 8, 157, 254, 222, 53, 0, 24, 0, 151, 35, 0, 230, 0, 66, 7, 118, 0, 177, 0, 109, 255, 227, 251, 212, 236, 0, 128, 228, 227, 225, 228, 141, 162, 247, 0, 1, 250, 159, 223, 166, 157, 159, 158, 198, 160, 171, 22, 252, 235, 2, 163, 241, 157, 234, 4, 31, 14, 156, 254, 207, 245, 165, 169, 161, 2, 166, 164, 251, 159, 157, 248, 162, 242, 4, 5, 227, 252, 161, 242, 7, 215, 158, 253, 157, 25, 235, 169, 228, 158, 252, 22, 31, 200, 223, 158, 154, 238, 226, 165, 235, 141, 155, 220, 244, 166, 165, 158, 170, 165, 160, 38, 157, 228, 231, 229, 219, 157, 17, 21, 205, 32, 244, 242, 239, 2, 160, 0, 13, 154, 255, 162, 165, 252, 239, 255, 227, 246, 0, 6, 15, 20, 212, 149, 0, 227, 29, 169, 40, 24, 234, 105, 65, 12, 0, 12, 217, 0, 129, 22, 221, 230, 235, 0, 24, 221, 172, 27, 29, 3, 205, 166, 46, 195, 146, 220, 7, 45, 164, 164, 144, 161, 200, 168, 233, 5, 165, 209, 255, 18,  255, 43, 228, 213, 250, 158, 250, 232, 207, 211, 146, 197, 42, 208, 169, 205, 244, 164) |

Масив 

Далі відбувається формування нових блоків, шляхом додавання до значень початкової популяції значень вектору швидкості за формулою (1.2). На цьому кроці також відбувається перевірка того, що значення входять до діапазону . Якщо значення більше або менше необхідного, воно береться за модулем розмірності блоку – 256.

Ще одним критерієм формування нових блоків є перевірка на бієктивність. Процедура перевірки описана в Алгоритмі 2.

Після даної процедури ми отримуємо нову популяцію розміру . До старої шляхом модернізації додалося ще  блоків і маємо:

|  |  |
| --- | --- |
| S-Box | NL |
| (99, 124, 119, 123, 242, 107, 111, 197, 48, 1, 103, 43, 254, 215, 171, 118,  202, 130, 201, 125, 250, 89, 71, 240, 173, 212, 162, 175, 156, 164, 114, 192,  183, 253, 147, 38, 54, 63, 247, 204, 52, 165, 229, 241, 113, 216, 49, 21,  4, 199, 35, 195, 24, 150, 5, 154, 7, 18, 128, 226, 235, 39, 178, 117,  9, 131, 44, 26, 27, 110, 90, 160, 82, 59, 214, 179, 41, 227, 47, 132,  83, 209, 0, 237, 32, 252, 177, 91, 106, 203, 190, 57, 74, 76, 88, 207,  208, 239, 170, 251, 67, 77, 51, 133, 69, 249, 2, 127, 80, 60, 159, 168,  81, 163, 64, 143, 146, 157, 56, 245, 188, 182, 218, 33, 16, 255, 243, 210,  205, 12, 19, 236, 95, 151, 68, 23, 196, 167, 126, 61, 100, 93, 25, 115,  96, 129, 79, 220, 34, 42, 144, 136, 70, 238, 184, 20, 222, 94, 11, 219,  224, 50, 58, 10, 73, 6, 36, 92, 194, 211, 172, 98, 145, 149, 228, 121,  231, 200, 55, 109, 141, 213, 78, 169, 108, 86, 244, 234, 101, 122, 174, 8,  186, 120, 37, 46, 28, 166, 180, 198, 232, 221, 116, 31, 75, 189, 139, 138,  112, 62, 181, 102, 72, 3, 246, 14, 97, 53, 87, 185, 134, 193, 29, 158,  225, 248, 152, 17, 105, 217, 142, 148, 155, 30, 135, 233, 206, 85, 40, 223,  140, 161, 137, 13, 191, 230, 66, 104, 65, 153, 45, 15, 176, 84, 187, 22) | 112 |
| (72, 67, 68, 107, 189, 235, 163, 243, 96, 102, 93, 224, 229, 101, 215, 152,  204, 222, 0, 131, 126, 244, 46, 193, 217, 64, 43, 187, 181, 228, 210, 221,  70, 6, 122, 164, 156, 39, 245, 172, 239, 237, 160, 192, 91, 167, 135, 9,  97, 207, 158, 50, 177, 203, 66, 145, 86, 194, 42, 254, 179, 240, 90, 38,  144, 208, 130, 117, 166, 139, 120, 59, 185, 53, 44, 115, 75, 150, 73, 143,  176, 116, 137, 227, 255, 129, 89, 47, 238, 3, 151, 242, 188, 209, 79, 231,  8, 214, 31, 200, 52, 87, 63, 230, 219, 11, 20, 213, 35, 170, 161, 108,  250, 132, 123, 212, 121, 140, 169, 36, 78, 196, 146, 25, 118, 232, 5, 28,  211, 220, 65, 112, 128, 45, 141, 119, 19, 32, 162, 92, 216, 69, 24, 218,  186, 54, 2, 76, 110, 241, 29, 56, 171, 14, 183, 147, 233, 182, 201, 103,  149, 81, 58, 246, 226, 82, 40, 17, 175, 159, 155, 114, 248, 251, 1, 125,  142, 74, 136, 197, 100, 133, 157, 109, 199, 94, 180, 198, 191, 80, 178, 16,  34, 234, 62, 12, 60, 48, 23, 27, 71, 85, 148, 124, 206, 168, 88, 225,  30, 22, 61, 165, 205, 154, 173, 13, 249, 247, 83, 98, 153, 104, 57, 49,  84, 4, 10, 77, 21, 252, 51, 33, 95, 202, 99, 37, 184, 18, 190, 195,  7, 138, 127, 134, 253, 236, 111, 174, 223, 55, 106, 113, 105, 26, 41, 15) | 94 |
| (242, 201, 159, 34, 75, 78, 207, 185, 225, 116, 94, 1, 140, 86, 232, 93,  209, 131, 165, 69, 177, 246, 194, 99, 171, 228, 70, 154, 199, 189, 80, 60,  250, 182, 144, 151, 241, 87, 100, 4, 183, 23, 252, 253, 25, 3, 179, 126,  68, 134, 195, 90, 218, 27, 57, 190, 222, 149, 143, 6, 184, 192, 198, 238,  55, 239, 114, 81, 254, 89, 36, 14, 37, 33, 56, 135, 213, 148, 203, 217,  124, 2, 88, 216, 113, 233, 255, 101, 146, 43, 142, 103, 188, 73, 175, 191,  123, 46, 84, 173, 247, 202, 214, 102, 138, 51, 166, 109, 119, 64, 145, 115,  244, 107, 163, 150, 193, 18, 221, 0, 136, 160, 17, 155, 62, 40, 223, 117,  49, 83, 229, 13, 118, 147, 54, 132, 7, 245, 226, 227, 63, 12, 161, 208,  9, 24, 121, 156, 48, 237, 157, 112, 235, 204, 15, 104, 71, 176, 67, 215,  219, 95, 120, 168, 79, 110, 251, 169, 22, 44, 122, 231, 197, 58, 76, 230,  139, 137, 97, 59, 128, 74, 186, 248, 11, 162, 66, 111, 133, 52, 19, 29,  42, 236, 98, 39, 10, 172, 181, 240, 45, 105, 206, 61, 16, 65, 85, 130,  205, 92, 47, 125, 167, 153, 180, 50, 243, 174, 127, 41, 234, 249, 152, 106,  28, 96, 196, 108, 38, 129, 5, 170, 26, 77, 178, 82, 31, 32, 20, 210,  21, 141, 91, 8, 158, 187, 72, 164, 211, 53, 35, 220, 30, 200, 224, 212) | 94 |
| (170, 185, 249, 145, 2, 13, 179, 208, 28, 196, 168, 80, 124, 112, 113, 243,  74, 238, 160, 65, 30, 251, 7, 33, 234, 161, 32, 61, 19, 206, 230, 165,  29, 131, 186, 236, 190, 0, 72, 106, 95, 96, 15, 193, 212, 98, 52, 53,  150, 26, 69, 36, 217, 205, 192, 246, 233, 56, 101, 232, 201, 58, 252, 1,  127, 121, 227, 68, 180, 31, 203, 91, 102, 147, 173, 167, 25, 250, 211, 221,  187, 47, 59, 88, 35, 129, 139, 235, 222, 132, 142, 105, 240, 120, 143, 34,  75, 245, 55, 116, 119, 93, 157, 27, 66, 8, 39, 18, 77, 89, 228, 4,  148, 51, 200, 14, 87, 130, 63, 141, 184, 151, 226, 152, 183, 43, 215, 163,  37, 254, 70, 46, 79, 117, 153, 242, 149, 138, 73, 100, 82, 45, 210, 97,  44, 162, 213, 40, 146, 241, 78, 83, 99, 41, 253, 92, 16, 209, 110, 64,  133, 189, 175, 174, 255, 54, 107, 118, 103, 22, 140, 123, 158, 42, 191, 49,  195, 84, 6, 9, 229, 21, 136, 60, 220, 94, 115, 48, 81, 90, 126, 111,  108, 216, 50, 197, 17, 198, 122, 20, 202, 207, 85, 231, 134, 24, 218, 128,  11, 166, 194, 38, 164, 199, 12, 177, 224, 3, 10, 172, 223, 156, 247, 239,  23, 181, 104, 182, 219, 188, 248, 214, 71, 155, 204, 144, 67, 57, 159, 237,  178, 5, 225, 76, 109, 169, 137, 135, 62, 125, 176, 86, 244, 154, 114, 171) | 92 |
| (240, 119, 114, 122, 237, 121, 117, 202, 54, 12, 102, 63, 252, 204, 176, 4,  203, 140, 180, 126, 150, 106, 69, 236, 178, 197, 154, 177, 159, 171, 125, 196, 172, 228, 145, 52, 65, 61, 247, 201, 72, 173, 222, 169, 111, 211, 59, 20,  14, 200, 48, 30, 41, 156, 87, 55, 16, 37, 43, 229, 230, 185, 24, 109,  175, 1, 219, 36, 42, 22, 94, 95, 93, 10, 67, 91, 45, 96, 50, 134,  164, 5, 15, 143, 56, 62, 168, 242, 120, 182, 186, 77, 86, 90, 88, 210,  187, 162, 194, 246, 66, 79, 53, 144, 85, 225, 141, 137, 76, 191, 160, 19,  99, 105, 71, 151, 216, 98, 68, 224, 234, 184, 118, 33, 27, 253, 155, 174,  206, 34, 223, 152, 199, 103, 74, 233, 179, 153, 130, 215, 113, 244, 25, 214, 80, 148, 131, 205, 157, 218, 132, 128, 81, 208, 92, 136, 167, 104, 31, 207,  217, 107, 58, 35, 110, 75, 142, 188, 193, 116, 70, 100, 18, 209, 82, 51,  8, 221, 64, 158, 139, 198, 83, 163, 129, 115, 238, 231, 249, 239, 166, 9,  29, 189, 40, 254, 32, 123, 6, 181, 243, 108, 49, 220, 89, 11, 124, 101,  112, 235, 165, 195, 13, 251, 57, 3, 84, 212, 127, 46, 135, 28, 17, 147,  60, 44, 138, 7, 97, 146, 133, 47, 149, 39, 73, 213, 226, 78, 2, 232,  190, 183, 245, 26, 23, 170, 227, 161, 0, 21, 250, 192, 248, 241, 38, 255) | 90 |
| (92, 85, 81, 101, 181, 211, 165, 235, 107, 98, 94, 190, 222, 55, 215, 144,  205, 207, 30, 125, 139, 236, 64, 186, 210, 90, 53, 183, 182, 221, 191, 203,  96, 40, 126, 156, 159, 46, 230, 157, 223, 206, 174, 202, 86, 153, 135, 22,  89, 199, 155, 63, 173, 177, 62, 151, 243, 179, 58, 227, 232, 224, 0, 161,  127, 18, 124, 108, 168, 133, 110, 61, 164, 52, 56, 121, 88, 41, 150, 220,  166, 128, 4, 38, 148, 75, 112, 111, 114, 20, 234, 217, 143, 187, 118, 226,  32, 198, 45, 201, 74, 99, 79, 42, 167, 218, 35, 197, 47, 131, 160, 113,  240, 132, 192, 219, 78, 129, 213, 71, 91, 83, 137, 9, 3, 214, 208, 49,  194, 116, 80, 109, 39, 69, 76, 115, 29, 82, 204, 105, 196, 65, 158, 67,  163, 1, 15, 93, 189, 229, 50, 120, 172, 48, 154, 216, 152, 97, 176, 122,  225, 26, 21, 209, 6, 11, 169, 17, 175, 146, 106, 25, 237, 84, 248, 13,  100, 138, 54, 178, 140, 245, 33, 104, 51, 68, 188, 28, 180, 170, 162, 16,  43, 251, 244, 72, 145, 242, 77, 103, 2, 117, 254, 119, 24, 228, 34, 27,  123, 195, 19, 252, 200, 60, 102, 253, 239, 134, 36, 171, 233, 14, 66, 37,  87, 57, 59, 185, 246, 238, 10, 241, 141, 130, 250, 7, 184, 149, 136, 12,  249, 247, 147, 95, 70, 231, 142, 255, 44, 73, 193, 212, 23, 5, 8, 31) | 94 |
| (221, 161, 159, 150, 81, 112, 207, 65, 109, 13, 56, 12, 144, 97, 232, 35,  191, 137, 96, 70, 166, 231, 117, 101, 173, 152, 208, 79, 178, 120, 130, 86,  250, 182, 222, 189, 225, 77, 106, 27, 202, 40, 226, 246, 50, 26, 158, 113,  73, 127, 95, 91, 216, 53, 69, 194, 211, 132, 138, 43, 46, 169, 203, 206,  60, 52, 122, 6, 233, 85, 57, 31, 47, 186, 78, 141, 184, 164, 195, 213,  129, 19, 80, 204, 170, 224, 239, 115, 171, 62, 145, 3, 187, 244, 167, 176,  123, 28, 87, 172, 223, 133, 199, 165, 196, 59, 143, 68, 131, 67, 155, 107,  124, 105, 162, 136, 180, 38, 242, 30, 217, 94, 51, 148, 72, 54, 237, 128,  125, 253, 200, 108, 102, 1, 66, 154, 34, 230, 205, 248, 84, 21, 157, 134,  18, 45, 252, 175, 58, 126, 214, 111, 156, 179, 116, 98, 75, 238, 245, 147,  103, 48, 247, 160, 41, 110, 251, 193, 22, 142, 9, 212, 190, 88, 0, 219,  99, 16, 135, 209, 140, 74, 100, 119, 15, 151, 234, 201, 24, 177, 82, 163,  235, 174, 90, 5, 7, 61, 39, 11, 188, 104, 89, 4, 32, 241, 198, 76,  240, 63, 71, 42, 2, 149, 49, 254, 17, 146, 197, 64, 227, 236, 139, 243,  33, 114, 23, 93, 44, 37, 218, 14, 121, 185, 153, 249, 228, 118, 183, 36,  20, 8, 215, 10, 220, 229, 92, 29, 255, 83, 55, 192, 25, 168, 210, 181) | 94 |
| (143, 230, 249, 227, 19, 159, 179, 63, 16, 162, 203, 70, 118, 107, 113, 100,  60, 204, 129, 61, 40, 171, 216, 42, 15, 79, 32, 190, 10, 59, 122, 105,  247, 52, 151, 200, 217, 4, 75, 101, 83, 91, 27, 176, 184, 95, 47, 46,  126, 34, 182, 198, 1, 2, 192, 14, 233, 207, 136, 232, 175, 58, 62, 8,  245, 121, 148, 68, 33, 188, 174, 142, 81, 127, 173, 116, 73, 221, 180, 193,  72, 21, 50, 88, 36, 123, 124, 202, 92, 195, 242, 153, 165, 117, 166, 56,  71, 224, 57, 23, 104, 114, 135, 31, 178, 22, 254, 49, 226, 78, 215, 67,  3, 53, 212, 231, 82, 25, 155, 133, 236, 137, 43, 74, 154, 39, 93, 149,  44, 213, 65, 134, 250, 234, 132, 28, 225, 169, 69, 51, 138, 26, 177, 243,  96, 144, 183, 98, 125, 223, 238, 181, 87, 228, 11, 45, 146, 17, 210, 130,  172, 161, 150, 147, 218, 99, 251, 12, 9, 139, 128, 109, 141, 112, 156, 237,  222, 64, 80, 48, 115, 199, 119, 102,191, 84, 54, 186, 160, 110, 170, 94,  108, 187, 106, 201, 38, 76, 211, 196, 167, 219, 85, 97, 248, 24, 7, 194,  246, 140, 77, 197, 0, 164, 252, 189, 253, 205, 35, 240, 235, 18, 89, 55,  30, 120, 157, 229, 163, 185, 86, 5, 37, 41, 208, 131, 29, 145, 206, 152,  6, 111, 255, 168, 103, 66, 239, 90, 220, 20, 209, 158, 241, 214, 244, 13) | 90 |

Масив 

Знову відбувається процедура сортування за нелінійністю популяції. Після сортування останні  блоків відкидаються і масив знову має розмір перед початком наступного кроку оптимізації.

|  |  |
| --- | --- |
| S-Box | NL |
| (99, 124, 119, 123, 242, 107, 111, 197, 48, 1, 103, 43, 254, 215, 171, 118,  202, 130, 201, 125, 250, 89, 71, 240, 173, 212, 162, 175, 156, 164, 114, 192,  183, 253, 147, 38, 54, 63, 247, 204, 52, 165, 229, 241, 113, 216, 49, 21,  4, 199, 35, 195, 24, 150, 5, 154, 7, 18, 128, 226, 235, 39, 178, 117,  9, 131, 44, 26, 27, 110, 90, 160, 82, 59, 214, 179, 41, 227, 47, 132,  83, 209, 0, 237, 32, 252, 177, 91, 106, 203, 190, 57, 74, 76, 88, 207,  208, 239, 170, 251, 67, 77, 51, 133, 69, 249, 2, 127, 80, 60, 159, 168,  81, 163, 64, 143, 146, 157, 56, 245, 188, 182, 218, 33, 16, 255, 243, 210,  205, 12, 19, 236, 95, 151, 68, 23, 196, 167, 126, 61, 100, 93, 25, 115,  96, 129, 79, 220, 34, 42, 144, 136, 70, 238, 184, 20, 222, 94, 11, 219,  224, 50, 58, 10, 73, 6, 36, 92, 194, 211, 172, 98, 145, 149, 228, 121,  231, 200, 55, 109, 141, 213, 78, 169, 108, 86, 244, 234, 101, 122, 174, 8,  186, 120, 37, 46, 28, 166, 180, 198, 232, 221, 116, 31, 75, 189, 139, 138,  112, 62, 181, 102, 72, 3, 246, 14, 97, 53, 87, 185, 134, 193, 29, 158,  225, 248, 152, 17, 105, 217, 142, 148, 155, 30, 135, 233, 206, 85, 40, 223,  140, 161, 137, 13, 191, 230, 66, 104, 65, 153, 45, 15, 176, 84, 187, 22) | 112 |
| (72, 67, 68, 107, 189, 235, 163, 243, 96, 102, 93, 224, 229, 101, 215, 152,  204, 222, 0, 131, 126, 244, 46, 193, 217, 64, 43, 187, 181, 228, 210, 221,  70, 6, 122, 164, 156, 39, 245, 172, 239, 237, 160, 192, 91, 167, 135, 9,  97, 207, 158, 50, 177, 203, 66, 145, 86, 194, 42, 254, 179, 240, 90, 38,  144, 208, 130, 117, 166, 139, 120, 59, 185, 53, 44, 115, 75, 150, 73, 143,  176, 116, 137, 227, 255, 129, 89, 47, 238, 3, 151, 242, 188, 209, 79, 231,  8, 214, 31, 200, 52, 87, 63, 230, 219, 11, 20, 213, 35, 170, 161, 108,  250, 132, 123, 212, 121, 140, 169, 36, 78, 196, 146, 25, 118, 232, 5, 28,  211, 220, 65, 112, 128, 45, 141, 119, 19, 32, 162, 92, 216, 69, 24, 218,  186, 54, 2, 76, 110, 241, 29, 56, 171, 14, 183, 147, 233, 182, 201, 103,  149, 81, 58, 246, 226, 82, 40, 17, 175, 159, 155, 114, 248, 251, 1, 125,  142, 74, 136, 197, 100, 133, 157, 109, 199, 94, 180, 198, 191, 80, 178, 16,  34, 234, 62, 12, 60, 48, 23, 27, 71, 85, 148, 124, 206, 168, 88, 225,  30, 22, 61, 165, 205, 154, 173, 13, 249, 247, 83, 98, 153, 104, 57, 49,  84, 4, 10, 77, 21, 252, 51, 33, 95, 202, 99, 37, 184, 18, 190, 195,  7, 138, 127, 134, 253, 236, 111, 174, 223, 55, 106, 113, 105, 26, 41, 15) | 94 |
| (242, 201, 159, 34, 75, 78, 207, 185, 225, 116, 94, 1, 140, 86, 232, 93,  209, 131, 165, 69, 177, 246, 194, 99, 171, 228, 70, 154, 199, 189, 80, 60,  250, 182, 144, 151, 241, 87, 100, 4, 183, 23, 252, 253, 25, 3, 179, 126,  68, 134, 195, 90, 218, 27, 57, 190, 222, 149, 143, 6, 184, 192, 198, 238,  55, 239, 114, 81, 254, 89, 36, 14, 37, 33, 56, 135, 213, 148, 203, 217,  124, 2, 88, 216, 113, 233, 255, 101, 146, 43, 142, 103, 188, 73, 175, 191,  123, 46, 84, 173, 247, 202, 214, 102, 138, 51, 166, 109, 119, 64, 145, 115,  244, 107, 163, 150, 193, 18, 221, 0, 136, 160, 17, 155, 62, 40, 223, 117,  49, 83, 229, 13, 118, 147, 54, 132, 7, 245, 226, 227, 63, 12, 161, 208,  9, 24, 121, 156, 48, 237, 157, 112, 235, 204, 15, 104, 71, 176, 67, 215,  219, 95, 120, 168, 79, 110, 251, 169, 22, 44, 122, 231, 197, 58, 76, 230,  139, 137, 97, 59, 128, 74, 186, 248, 11, 162, 66, 111, 133, 52, 19, 29,  42, 236, 98, 39, 10, 172, 181, 240, 45, 105, 206, 61, 16, 65, 85, 130,  205, 92, 47, 125, 167, 153, 180, 50, 243, 174, 127, 41, 234, 249, 152, 106,  28, 96, 196, 108, 38, 129, 5, 170, 26, 77, 178, 82, 31, 32, 20, 210,  21, 141, 91, 8, 158, 187, 72, 164, 211, 53, 35, 220, 30, 200, 224, 212) | 94 |
| (92, 85, 81, 101, 181, 211, 165, 235, 107, 98, 94, 190, 222, 55, 215, 144,  205, 207, 30, 125, 139, 236, 64, 186, 210, 90, 53, 183, 182, 221, 191, 203,  96, 40, 126, 156, 159, 46, 230, 157, 223, 206, 174, 202, 86, 153, 135, 22,  89, 199, 155, 63, 173, 177, 62, 151, 243, 179, 58, 227, 232, 224, 0, 161,  127, 18, 124, 108, 168, 133, 110, 61, 164, 52, 56, 121, 88, 41, 150, 220,  166, 128, 4, 38, 148, 75, 112, 111, 114, 20, 234, 217, 143, 187, 118, 226,  32, 198, 45, 201, 74, 99, 79, 42, 167, 218, 35, 197, 47, 131, 160, 113,  240, 132, 192, 219, 78, 129, 213, 71, 91, 83, 137, 9, 3, 214, 208, 49,  194, 116, 80, 109, 39, 69, 76, 115, 29, 82, 204, 105, 196, 65, 158, 67,  163, 1, 15, 93, 189, 229, 50, 120, 172, 48, 154, 216, 152, 97, 176, 122,  225, 26, 21, 209, 6, 11, 169, 17, 175, 146, 106, 25, 237, 84, 248, 13,  100, 138, 54, 178, 140, 245, 33, 104, 51, 68, 188, 28, 180, 170, 162, 16,  43, 251, 244, 72, 145, 242, 77, 103, 2, 117, 254, 119, 24, 228, 34, 27,  123, 195, 19, 252, 200, 60, 102, 253, 239, 134, 36, 171, 233, 14, 66, 37,  87, 57, 59, 185, 246, 238, 10, 241, 141, 130, 250, 7, 184, 149, 136, 12,  249, 247, 147, 95, 70, 231, 142, 255, 44, 73, 193, 212, 23, 5, 8, 31) | 94 |

Масив 

Отримали:

|  |  |
| --- | --- |
| Нелінійності початкової популяції | Нелінійності нової популяції |
| 112 | 112 |
| 94 | 94 |
| 94 | 94 |
| 92 | 94 |

Нелінійності популяцій

Далі всі процедури повторюються, доти, доки не закінчиться задана кількість ітерацій, або доки не буде знайдено необхідний блок.

Модифікований алгоритм працює майже так само, тільки на перший крок оптимізації починається з перемішування першого блоку в популяції, щоб не сильно погіршити одні його показники та покращити інші і на його основі вже далі проводити оптимізацію.

Припустимо, як і в минулому прикладі, у нас на початку першого кроку генерується популяція:

|  |  |
| --- | --- |
| S-Box | NL |
| (99, 124, 119, 123, 242, 107, 111, 197, 48, 1, 103, 43, 254, 215, 171, 118,  202, 130, 201, 125, 250, 89, 71, 240, 173, 212, 162, 175, 156, 164, 114, 192,  183, 253, 147, 38, 54, 63, 247, 204, 52, 165, 229, 241, 113, 216, 49, 21,  4, 199, 35, 195, 24, 150, 5, 154, 7, 18, 128, 226, 235, 39, 178, 117,  9, 131, 44, 26, 27, 110, 90, 160, 82, 59, 214, 179, 41, 227, 47, 132,  83, 209, 0, 237, 32, 252, 177, 91, 106, 203, 190, 57, 74, 76, 88, 207,  208, 239, 170, 251, 67, 77, 51, 133, 69, 249, 2, 127, 80, 60, 159, 168,  81, 163, 64, 143, 146, 157, 56, 245, 188, 182, 218, 33, 16, 255, 243, 210,  205, 12, 19, 236, 95, 151, 68, 23, 196, 167, 126, 61, 100, 93, 25, 115,  96, 129, 79, 220, 34, 42, 144, 136, 70, 238, 184, 20, 222, 94, 11, 219,  224, 50, 58, 10, 73, 6, 36, 92, 194, 211, 172, 98, 145, 149, 228, 121,  231, 200, 55, 109, 141, 213, 78, 169, 108, 86, 244, 234, 101, 122, 174, 8,  186, 120, 37, 46, 28, 166, 180, 198, 232, 221, 116, 31, 75, 189, 139, 138,  112, 62, 181, 102, 72, 3, 246, 14, 97, 53, 87, 185, 134, 193, 29, 158,  225, 248, 152, 17, 105, 217, 142, 148, 155, 30, 135, 233, 206, 85, 40, 223,  140, 161, 137, 13, 191, 230, 66, 104, 65, 153, 45, 15, 176, 84, 187, 22) | 112 |
| (72, 67, 68, 107, 189, 235, 163, 243, 96, 102, 93, 224, 229, 101, 215, 152,  204, 222, 0, 131, 126, 244, 46, 193, 217, 64, 43, 187, 181, 228, 210, 221,  70, 6, 122, 164, 156, 39, 245, 172, 239, 237, 160, 192, 91, 167, 135, 9,  97, 207, 158, 50, 177, 203, 66, 145, 86, 194, 42, 254, 179, 240, 90, 38,  144, 208, 130, 117, 166, 139, 120, 59, 185, 53, 44, 115, 75, 150, 73, 143,  176, 116, 137, 227, 255, 129, 89, 47, 238, 3, 151, 242, 188, 209, 79, 231,  8, 214, 31, 200, 52, 87, 63, 230, 219, 11, 20, 213, 35, 170, 161, 108,  250, 132, 123, 212, 121, 140, 169, 36, 78, 196, 146, 25, 118, 232, 5, 28,  211, 220, 65, 112, 128, 45, 141, 119, 19, 32, 162, 92, 216, 69, 24, 218,  186, 54, 2, 76, 110, 241, 29, 56, 171, 14, 183, 147, 233, 182, 201, 103,  149, 81, 58, 246, 226, 82, 40, 17, 175, 159, 155, 114, 248, 251, 1, 125,  142, 74, 136, 197, 100, 133, 157, 109, 199, 94, 180, 198, 191, 80, 178, 16,  34, 234, 62, 12, 60, 48, 23, 27, 71, 85, 148, 124, 206, 168, 88, 225,  30, 22, 61, 165, 205, 154, 173, 13, 249, 247, 83, 98, 153, 104, 57, 49,  84, 4, 10, 77, 21, 252, 51, 33, 95, 202, 99, 37, 184, 18, 190, 195,  7, 138, 127, 134, 253, 236, 111, 174, 223, 55, 106, 113, 105, 26, 41, 15) | 94 |
| (242, 201, 159, 34, 75, 78, 207, 185, 225, 116, 94, 1, 140, 86, 232, 93,  209, 131, 165, 69, 177, 246, 194, 99, 171, 228, 70, 154, 199, 189, 80, 60,  250, 182, 144, 151, 241, 87, 100, 4, 183, 23, 252, 253, 25, 3, 179, 126,  68, 134, 195, 90, 218, 27, 57, 190, 222, 149, 143, 6, 184, 192, 198, 238,  55, 239, 114, 81, 254, 89, 36, 14, 37, 33, 56, 135, 213, 148, 203, 217,  124, 2, 88, 216, 113, 233, 255, 101, 146, 43, 142, 103, 188, 73, 175, 191,  123, 46, 84, 173, 247, 202, 214, 102, 138, 51, 166, 109, 119, 64, 145, 115,  244, 107, 163, 150, 193, 18, 221, 0, 136, 160, 17, 155, 62, 40, 223, 117,  49, 83, 229, 13, 118, 147, 54, 132, 7, 245, 226, 227, 63, 12, 161, 208,  9, 24, 121, 156, 48, 237, 157, 112, 235, 204, 15, 104, 71, 176, 67, 215,  219, 95, 120, 168, 79, 110, 251, 169, 22, 44, 122, 231, 197, 58, 76, 230,  139, 137, 97, 59, 128, 74, 186, 248, 11, 162, 66, 111, 133, 52, 19, 29,  42, 236, 98, 39, 10, 172, 181, 240, 45, 105, 206, 61, 16, 65, 85, 130,  205, 92, 47, 125, 167, 153, 180, 50, 243, 174, 127, 41, 234, 249, 152, 106,  28, 96, 196, 108, 38, 129, 5, 170, 26, 77, 178, 82, 31, 32, 20, 210,  21, 141, 91, 8, 158, 187, 72, 164, 211, 53, 35, 220, 30, 200, 224, 212) | 94 |
| (170, 185, 249, 145, 2, 13, 179, 208, 28, 196, 168, 80, 124, 112, 113, 243,  74, 238, 160, 65, 30, 251, 7, 33, 234, 161, 32, 61, 19, 206, 230, 165,  29, 131, 186, 236, 190, 0, 72, 106, 95, 96, 15, 193, 212, 98, 52, 53,  150, 26, 69, 36, 217, 205, 192, 246, 233, 56, 101, 232, 201, 58, 252, 1,  127, 121, 227, 68, 180, 31, 203, 91, 102, 147, 173, 167, 25, 250, 211, 221,  187, 47, 59, 88, 35, 129, 139, 235, 222, 132, 142, 105, 240, 120, 143, 34,  75, 245, 55, 116, 119, 93, 157, 27, 66, 8, 39, 18, 77, 89, 228, 4,  148, 51, 200, 14, 87, 130, 63, 141, 184, 151, 226, 152, 183, 43, 215, 163,  37, 254, 70, 46, 79, 117, 153, 242, 149, 138, 73, 100, 82, 45, 210, 97,  44, 162, 213, 40, 146, 241, 78, 83, 99, 41, 253, 92, 16, 209, 110, 64,  133, 189, 175, 174, 255, 54, 107, 118, 103, 22, 140, 123, 158, 42, 191, 49,  195, 84, 6, 9, 229, 21, 136, 60, 220, 94, 115, 48, 81, 90, 126, 111,  108, 216, 50, 197, 17, 198, 122, 20, 202, 207, 85, 231, 134, 24, 218, 128,  11, 166, 194, 38, 164, 199, 12, 177, 224, 3, 10, 172, 223, 156, 247, 239,  23, 181, 104, 182, 219, 188, 248, 214, 71, 155, 204, 144, 67, 57, 159, 237,  178, 5, 225, 76, 109, 169, 137, 135, 62, 125, 176, 86, 244, 154, 114, 171) | 92 |

Масив 

Після цього невелика кількість значень з найкращого блоку перемішуються, щоб погіршити нелінійність не дуже значним чином. Отримуємо:

|  |  |
| --- | --- |
| S-Box | NL |
| (99, 124, 119, 123, 242, 107, 111, 197, 48, 1, 103, 43, 254, 215, 171, 118,  202, 130, 201, 125, 250, 89, 71, 240, 173, 212, 162, 175, 156, 164, 114, 192,  183, 253, 147, 38, 54, 63, 247, 204, 52, 165, 229, 241, 113, 216, 49, 21,  4, 199, 35, 195, 24, 150, 5, 154, 7, 18, 128, 226, 235, 39, 178, 117,  9, 131, 44, 26, 27, 110, 90, 160, 82, 59, 214, 179, 41, 227, 47, 132,  83, 209, 0, 237, 32, 252, 177, 91, 106, 203, 190, 57, 74, 76, 88, 207,  208, 239, 170, 251, 67, 77, 51, 133, 69, 249, 2, 127, 80, 60, 159, 168,  81, 163, 64, 143, 146, 157, 56, 245, 188, 182, 218, 33, 16, 255, 243, 210,  205, 12, 19, 236, 95, 151, 68, 23, 196, 167, 126, 61, 100, 93, 25, 115,  96, 129, 79, 220, 34, 42, 144, 136, 70, 238, 184, 20, 222, 94, 11, 219,  224, 50, 58, 10, 73, 6, 36, 92, 194, 211, 172, 98, 145, 149, 228, 121,  231, 200, 55, 109, 141, 213, 78, 169, 108, 86, 244, 234, 101, 122, 174, 8,  186, 120, 37, 46, 28, 166, 180, 198, 232, 221, 116, 31, 75, 189, 139, 138,  112, 62, 181, 102, 72, 3, 246, 14, 97, 53, 87, 185, 134, 193, 29, 158,  225, 248, 152, 17, 105, 217, 142, 148, 155, 30, 135, 233, 206, 85, 40, 223,  140, 161, 137, 13, 191, 230, 66, 104, 65, 153, 45, 15, 176, 84, 187, 22) | 112 |
| (99, 124, 119, 123, 242, 107, 111, 197, 48, 1, 103, 43, 254, 215, 171, 157,  202, 130, 201, 125, 250, 89, 71, 240, 173, 212, 162, 175, 156, 164, 114, 192, 183, 253, 147, 38, 54, 63, 72, 77, 52, 165, 229, 241, 113, 216, 49, 131,  4, 199, 35, 195, 24, 150, 5, 154, 7, 18, 128, 226, 235, 39, 178, 117,  9, 21, 44, 26, 27, 110, 90, 160, 82, 59, 214, 179, 41, 227, 47, 132,  83, 209, 0, 237, 32, 252, 177, 91, 106, 203, 190, 57, 74, 76, 88, 207,  208, 239, 170, 251, 67, 56, 51, 133, 69, 249, 2, 127, 80, 60, 159, 168,  81, 163, 64, 143, 146, 118, 86, 245, 188, 182, 218, 33, 16, 255, 243, 210,  205, 12, 19, 236, 95, 151, 68, 23, 196, 167, 126, 61, 100, 93, 25, 115,  96, 129, 79, 220, 34, 42, 144, 136, 70, 238, 184, 20, 222, 94, 11, 219,  224, 50, 58, 10, 73, 6, 36, 92, 194, 211, 172, 98, 145, 149, 228, 121,  231, 200, 55, 109, 141, 213, 78, 169, 108, 62, 244, 234, 101, 122, 174, 8,  186, 120, 37, 46, 28, 166, 180, 198, 232, 221, 116, 31, 75, 189, 139, 138,  112, 148, 181, 102, 191, 3, 246, 14, 97, 53, 87, 185, 134, 193, 29, 158,  225, 248, 152, 17, 105, 217, 142, 15, 155, 30, 135, 233, 206, 85, 40, 223,  140, 161, 137, 13, 176, 230, 66, 104, 65, 153, 45, 22, 247, 84, 187, 204) | 106 |
| (242, 201, 159, 34, 75, 78, 207, 185, 225, 116, 94, 1, 140, 86, 232, 93,  209, 131, 165, 69, 177, 246, 194, 99, 171, 228, 70, 154, 199, 189, 80, 60,  250, 182, 144, 151, 241, 87, 100, 4, 183, 23, 252, 253, 25, 3, 179, 126,  68, 134, 195, 90, 218, 27, 57, 190, 222, 149, 143, 6, 184, 192, 198, 238,  55, 239, 114, 81, 254, 89, 36, 14, 37, 33, 56, 135, 213, 148, 203, 217,  124, 2, 88, 216, 113, 233, 255, 101, 146, 43, 142, 103, 188, 73, 175, 191,  123, 46, 84, 173, 247, 202, 214, 102, 138, 51, 166, 109, 119, 64, 145, 115,  244, 107, 163, 150, 193, 18, 221, 0, 136, 160, 17, 155, 62, 40, 223, 117,  49, 83, 229, 13, 118, 147, 54, 132, 7, 245, 226, 227, 63, 12, 161, 208,  9, 24, 121, 156, 48, 237, 157, 112, 235, 204, 15, 104, 71, 176, 67, 215,  219, 95, 120, 168, 79, 110, 251, 169, 22, 44, 122, 231, 197, 58, 76, 230,  139, 137, 97, 59, 128, 74, 186, 248, 11, 162, 66, 111, 133, 52, 19, 29,  42, 236, 98, 39, 10, 172, 181, 240, 45, 105, 206, 61, 16, 65, 85, 130,  205, 92, 47, 125, 167, 153, 180, 50, 243, 174, 127, 41, 234, 249, 152, 106,  28, 96, 196, 108, 38, 129, 5, 170, 26, 77, 178, 82, 31, 32, 20, 210,  21, 141, 91, 8, 158, 187, 72, 164, 211, 53, 35, 220, 30, 200, 224, 212) | 94 |
| (170, 185, 249, 145, 2, 13, 179, 208, 28, 196, 168, 80, 124, 112, 113, 243,  74, 238, 160, 65, 30, 251, 7, 33, 234, 161, 32, 61, 19, 206, 230, 165,  29, 131, 186, 236, 190, 0, 72, 106, 95, 96, 15, 193, 212, 98, 52, 53,  150, 26, 69, 36, 217, 205, 192, 246, 233, 56, 101, 232, 201, 58, 252, 1,  127, 121, 227, 68, 180, 31, 203, 91, 102, 147, 173, 167, 25, 250, 211, 221,  187, 47, 59, 88, 35, 129, 139, 235, 222, 132, 142, 105, 240, 120, 143, 34,  75, 245, 55, 116, 119, 93, 157, 27, 66, 8, 39, 18, 77, 89, 228, 4,  148, 51, 200, 14, 87, 130, 63, 141, 184, 151, 226, 152, 183, 43, 215, 163,  37, 254, 70, 46, 79, 117, 153, 242, 149, 138, 73, 100, 82, 45, 210, 97,  44, 162, 213, 40, 146, 241, 78, 83, 99, 41, 253, 92, 16, 209, 110, 64,  133, 189, 175, 174, 255, 54, 107, 118, 103, 22, 140, 123, 158, 42, 191, 49,  195, 84, 6, 9, 229, 21, 136, 60, 220, 94, 115, 48, 81, 90, 126, 111,  108, 216, 50, 197, 17, 198, 122, 20, 202, 207, 85, 231, 134, 24, 218, 128,  11, 166, 194, 38, 164, 199, 12, 177, 224, 3, 10, 172, 223, 156, 247, 239,  23, 181, 104, 182, 219, 188, 248, 214, 71, 155, 204, 144, 67, 57, 159, 237,  178, 5, 225, 76, 109, 169, 137, 135, 62, 125, 176, 86, 244, 154, 114, 171) | 92 |

Масив 

Як вже було сказано раніше отриманий таким чином блок має гарну нелінійність, але всі інші параметри є поганими, тому далі продовжується оптимізація популяції. Всі наступні ітерації не відрізняються від алгоритму зі статті. В нашому випадку алгоритм здатний знаходити S-Box’и з нелінійністю 104 та іншими показниками, які задовольняють умовам захищеності. При цьому відмінність від AES-Sbox’у досягає 30-40 позицій у блоці.

Дані експериментальних запусків з різною цільовою нелінійністю, статистика знаходження потрібних блоків, а також самі блоки буде наведено нижче.