

## **ПРИМЕНЕНИЕ КОДА ГРЕЯ В ГЕНЕТИЧЕСКОМ АЛГОРИТМЕ ПРИ КОДИРОВАНИИ ПРИЗНАКОВ, ПРЕДСТАВЛЯЕМЫХ ЦЕЛЫМИ ЧИСЛАМИ**

**Архипов А.Н., Панов А.В.**

*МИРЭА - Российский технологический университет, 119454, Россия, г. Москва, проспект Вернадского, 78,  
e-mail: arhipov2605@gmail.com*

---

**Генетические алгоритмы применяются для решения поисковых задач и задач оптимизации. Важной аспектом при его проектировании является определение способа представления хромосомы особи в популяции. Рассматривается подход, применяемый при представлении целочисленных решений с использованием кода Грея и сравнение его с классическим бинарным представлением.**

---

Ключевые слова: генетический алгоритм, хромосома, мутация, кроссовер, код Грея.

## **APPLICATION OF GRAY CODE IN GENETIC ALGORITHM FOR REPRESENTATION OF ATTRIBUTES REPRESENTED AS INTEGER NUMBERS**

**Arkhipov A.N., Panov A.V.**

*MIREA - Russian Technological University, 119454, Moscow, 78 Vernadskogo Avenue, Russia,  
e-mail: arhipov2605@gmail.com*

---

**Genetic algorithms are used to solve various search and optimization problems. An important part in its design is the way in which the chromosome is represented. Analyzing the approach taken in presenting integer solutions using Gray code and comparing it with the classical binary representation.**

---

Key words: genetic algorithm, chromosome, mutation, crossover, Gray code.

### **Введение**

Генетический алгоритм является средством решения задач таких, как оптимизация поисковых алгоритмов и моделирование различных процессов с использованием механизмов, работающих согласно естественному отбору в природе. [1]

В ходе своей работы генетический алгоритм использует операторы выборки, скрещивания и мутации [2], итеративное выполнение которых приводит к нахождению оптимального решения. Эти операции выполняются над популяцией особей, каждая из которых представляется в виде хромосомы. Способ представления хромосомы зависит от решаемой задачи. При необходимости поиска численных решений возможным способом представления является массив нулей и единиц, соответствующий двоичному представлению числа. [5]

### **Задачи поиска целочисленного решения**

Поиск целочисленных решений применяется для решения различных задач. Стандартным примером является нахождение экстремума функции на заданном участке. Хотя существуют и другие способы решения этой задачи, она часто применяется при описании подхода генетического алгоритма ввиду своей простоты.

Аналогично поиску целочисленного решения в заданном диапазоне чисел, возможно кодирование дробного решения от 0 до 1 битовым массивом, где дробный результат может быть найден, как отношение числа, полученного при переходе из бинарного представления в десятичное, к максимальному значению, которое может быть задано битовой последовательностью указанной длины. Количество битов в таком случае определит точность получаемого результата.

Также хромосома может представлять набор целых чисел. В таком случае битовый массив будет иметь длину, достаточную для последовательной записи бинарного представления необходимого количества чисел. Например, в случае, когда решение представляется двумя целыми числами в диапазоне от 0 до 1023, хромосома будет иметь 20 битов – по 10 для каждого числа.

Использование бинарного представления десятичного числа позволяет однозначно определить способ конвертации хромосомы особи в решение задачи, но имеет один существенный недостаток – изменение одной позиции в битовом массиве может существенно изменить численное представление. Это связано с тем, что в двоичном представлении десятичного числа каждая позиция имеет разное вес в целочисленном значении в зависимости от ее порядкового номера.

Генетический алгоритм реализуется следующим набором операций, которые последовательно выполняются в цикле: скрещивание, мутация, вычисление значения целевой функции для особей в поколении, формирование нового поколения посредством селекции и проверка условия останова. Генетические операторы модифицируют текущие значения хромосом в популяции.

Особенность представления десятичных чисел при помощи их перевода в двоичное представление, заключающаяся в частых и существенных изменениях, имеет значение на двух этапах работы генетического алгоритма – этапе кроссовера и этапе мутации. В процессе кроссовера дочерняя хромосома должна наследовать родительские характеристики наиболее точно.

Особое влияние способ представления хромосомы играет в ходе этапа мутации, на котором изменения генотипа должны сказываться на искомым значениях минимально. Это важно, поскольку частые и большие изменения итогового значения делают процесс схождения более хаотичным, нивелируя одно из основных преимуществ генетического алгоритма, заключающееся в постепенном схождении к искомому решению. Резкие изменения добавляют излишнюю случайности в работу алгоритма.

Решением этой проблемы является изменение способа преобразования представления хромосомы битовым массивом в число. Подход, часто применяемый при реализации генетического алгоритма, который позволяет частично решить описанную выше проблему – использование кода Грея. [4]

#### **Применение кода Грея в генетических алгоритмах**

Кодирование битового представления при помощи кода Грея обладает важным для работы генетического алгоритма свойством – смежные числа в коде Грея всегда отличаются на один бит. [7] Это обеспечивает то, что соседние десятичные числа будут соседними после конвертации в битовый массив.

Существуют различные способы построения кодов Грея заданной длины, при их использовании в генетическом алгоритме важно придерживаться единого способа конвертации. Часто применяется рефлексивный двоичный код Грея, в котором первая половина значений при изменении порядка эквивалентна второй половине, за исключением старшего бита, а старший бит инвертируется. [4]

Способ генерации кодов Грея, использованный в рамках данной работы, реализованный в виде кода на языке C#, приведен в листинге 1.

Листинг 1. Генерация кодов Грея.

```
public static int[] ConvertBinaryToGray(int[] binaryArray)
{
    // объявление массива соответствующей длины
    int[] grayArray = new int[binaryArray.Length];

    // бит старшего порядка копируется из бинарного представления
    grayArray[grayArray.Length - 1] = binaryArray[binaryArray.Length - 1];
    for (int i = grayArray.Length - 2; i >= 0; i = i - 1)
    {
        // оставшиеся биты определяются попарным сравнением двух битов бинарного представления:
        // при одинаковом значении в коде Грея ставится 0; при различном - ставится 1
        if (binaryArray[i] == binaryArray[i + 1])
        {
            grayArray[i] = 0;
        }
        else
        {
            grayArray[i] = 1;
        }
    }
    return grayArray;
}
```

Такой способ является универсальным для генерации кодов Грея любой заданной длины битового представления, которая в примере задается константой `ArrayLength`. Он заключается в копировании старшего бита из бинарного представления и определения оставшихся битов путем попарного сравнения двух соседних битов двоичного представления. В коде Грея ставится 0 при совпадении двух соответствующих битов бинарного представления и 1 – в случае, когда они отличаются.

В таблице 1 представлено сравнение бинарного представления и кодов Грея, полученных способом, приведенным выше, для первых 16 десятичных чисел в виде битовых массивов.

Таблица 1. Сравнение бинарного представления и кода Грея

Десятичное число	Двоичное кодирование	Код Грея
0	[0,0,0,0]	[0,0,0,0]
1	[0,0,0,1]	[0,0,0,1]
2	[0,0,1,0]	[0,0,1,1]
3	[0,0,1,1]	[0,0,1,0]
4	[0,1,0,0]	[0,1,1,0]
5	[0,1,0,1]	[0,1,1,1]
6	[0,1,1,0]	[0,1,0,1]
7	[0,1,1,1]	[0,1,0,0]
8	[1,0,0,0]	[1,1,0,0]
9	[1,0,0,1]	[1,1,0,1]
10	[1,0,1,0]	[1,1,1,1]
11	[1,0,1,1]	[1,1,1,0]
12	[1,1,0,0]	[1,0,1,0]
13	[1,1,0,1]	[1,0,1,1]
14	[1,1,1,0]	[1,0,0,1]
15	[1,1,1,1]	[1,0,0,0]

В приведенной таблице видно, что соседние числа в десятичном представлении являются соседними и в коде Грея, но не в двоичном представлении. Особую роль это играет на этапе мутации [3] – изменение одного бита с наибольшей вероятностью приведет к минимальному или незначительному изменению. В то же время есть небольшая вероятность значительного изменения, большего, чем в бинарном представлении. Но такие мутации случаются относительно редко, а особи, мутировавшие подобным образом, с высокой вероятностью будут исключены на следующих этапах селекции в случае, если мутация негативно повлияла на приспособленность особи.

Представление кодом Грея возможно на любом диапазоне чисел и определяется длиной битовой последовательности. Аналогично бинарному представлению, коды Грея определенной длины позволяют представить интервал чисел от 0 до  $2^n - 1$ , где  $n$  – количество битов.

#### Сравнение бинарного представления и кода Грея

Все возможные битовые массивы длиной в восемь битов дадут числа от 0 до 255 при конвертации в десятичное число. Каждая возможная последовательность имеет одно десятичное число, сопоставимое с ней, при этом конвертация из кода Грея даст другой результат, нежели конвертация из бинарного представления. [8]

При сравнении того, как сильно может измениться значение битового массива при конвертации различными способами, возьмем все возможные битовые представления чисел от 0 до 255 и проведем все возможные мутации с ними. Каждая мутация представляет собой изменение значения одного бита. Таким образом, получим набор изменений значения для 256 чисел, по 8 на каждое.

Например, при мутации хромосомы [0,0,1,1,1,0,1,0] в [0,0,1,0,1,0,1,0] при изменении четвертого бита, в бинарном представлении произойдет изменение числа 92 в 83, а в коде Грея число 104 превратится в 103. Изменением для первого случая при этом будет разность между 92 и 83, то есть 9. Для второго случая изменение минимально и составляет единицу.

Проведя все 2048 измерений для бинарного представления и такое же количество для конвертации с использованием кода Грея, можно судить о частоте значительных и незначительных изменений при мутации, которая является определяющим фактором применения кода Грея при реализации генетических алгоритмов.

На рис.1 приведено распределение переходов при мутации в одном бите для обозначенного диапазона битовых массивов, использующих код Грея при конвертации.

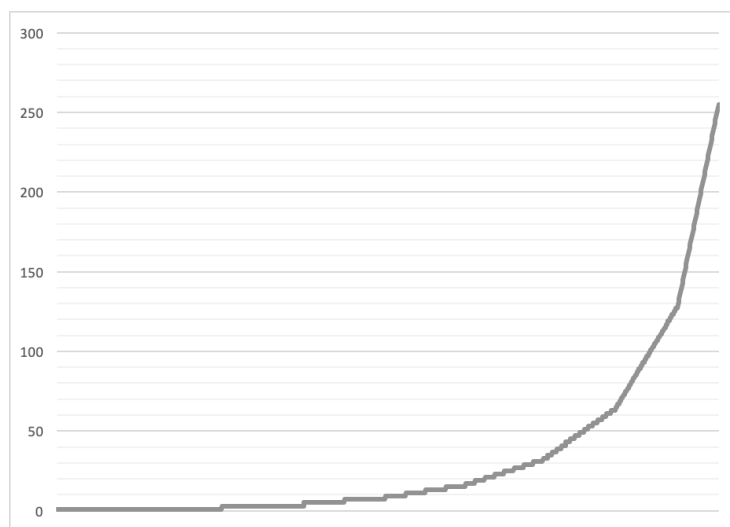


Рис. 1. Динамика изменений при мутации с использованием кода Грея

На рис. 2 представлено распределение переходов для аналогичного набора хромосом, конвертируемых при помощи бинарного представления десятичных чисел.

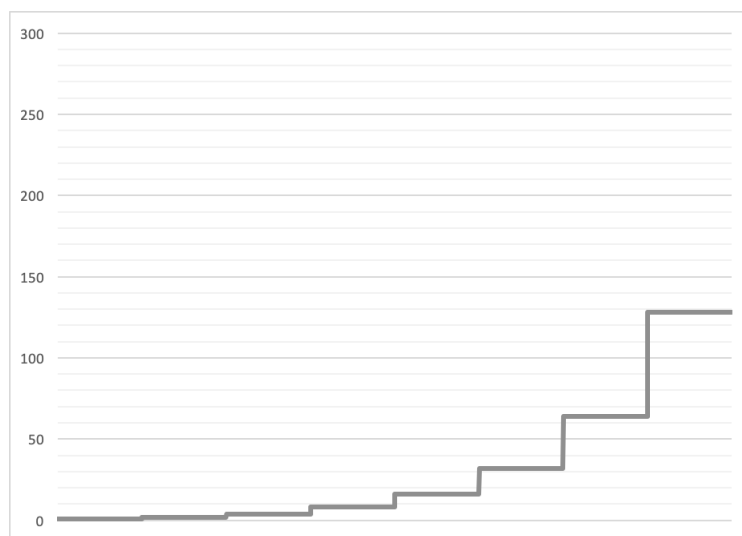


Рис. 2. Динамика изменений при мутации с использованием бинарного представления

На приведенных графиках видно подтверждение описанного выше свойства кода Грея, большая часть мутаций приводит к небольшому изменению численного представления. Изменения при использовании кода Грея растут медленнее. Подобные пропорции и форма графиков сохраняются при любой выбранной длине битового массива и, соответственно, диапазона десятичных чисел.

Видно, что использование бинарного кода ограничивает максимально возможное изменение числом 128, то есть половиной диапазона всех чисел. В то же время код Грея допускает существенные изменения, встречаются существенные изменения. Например, число 255, представляемое в коде Грея как  $[1,0,0,0,0,0,0]$  может мутировать в число 0, если будет изменено значение крайнего левого бита и хромосома примет вид  $[0,0,0,0,0,0,0]$ .

Несмотря на этот негативный момент, важно отметить, что максимально возможное изменение в бинарном представлении произойдет один раз из восьми. При использовании кода Грея максимальное изменение возможно лишь при одной мутации, хотя его абсолютное значение и выше. Код Грея обеспечивает меньшую вероятность

большого изменения при изменении одного бита, большинство изменений будут приводить к несущественному изменению.

Представление хромосомы не является единственным параметром, влияющим на результат работы генетического алгоритма. Поиск глобального оптимального решения зависит от вида функции, максимум которой ищется и количества локальных экстремумов [6], вероятностей кроссовера и мутации, установленных разработчиком, размера популяции и др. Поэтому использование бинарного представления может привести к лучшей работе алгоритма в отдельно взятых случаях и может быть применено для сравнения результатов работы и получения более широкого представления о структуре решаемой проблемы.

#### **Заключение**

Применение кода Грея при реализации генетического алгоритма при поиске целочисленных решений обусловлено его особенностью минимизировать эффект оператора мутации. Выбор представления хромосомы зависит от решаемой задачи, но применение кода Грея является предпочтительным в общем случае и ведет к большей вероятности быстрого схождения алгоритма к правильному решению. На практике, при решении комплексной задачи с большим количеством переменных оба способа могут быть использованы отдельно друг от друга с последующим сравнением результатов.

#### **Список литературы**

---

1. Гладков, Л.А., Курейчик, В.В., Курейчик, В.М. Генетические алгоритмы / Под ред. В.М. Курейчика. – 2-е изд., испр. и доп. – М.: Физматлит, 2006. – 320 с.
2. Генетический алгоритм. Стандарт: Ч. 1: Описание стандартного генетического алгоритма (сГА) / А. Б. Сергиенко, П. В. Галушин, В. В. Бухтояров [и др.]; Сибирский гос. аэрокосмический ун-т им. М. Ф. Решетнева. – Красноярск, 2010. - 384 с.
3. Рутковская, Д., Пилинский, М., Рутковский, Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. – 2-е изд., стереотип. – М.: Горячая линия– Телеком, 2013. – 384 с.
4. Код Грея - Википедия [Электронный ресурс]. — Режим доступа: [https://ru.wikipedia.org/wiki/Код\\_Грея](https://ru.wikipedia.org/wiki/Код_Грея) (дата обращения: 04.04.21).
5. M. Wall. Galib: A C++ library of genetic algorithm components. Mechanical Engineering Department, Massachusetts Institute of Technology, 87, 1996.
6. Whitley, D (1999): A free lunch proof for Gray versus binary encodings, Proc. Genetic and Evolutionary Computation (GECCO-1999), pp. 726-733.
7. Gvozdzak P. Goddyn L. Binary gray codes with long bit runs. Electr. J. Comb, 10:1–10, 2003.
8. Rana, S. and Whitley, D. (1997). Bit Representations with a Twist. In Ba ĉk, T., editor, Proceedings of the Seventh International Conference on Genetic Algorithms, pages 188– 195. Morgan Kaufmann.

#### **References**

---

1. Gladkov L.A., Kurejchik V.V., Kurejchik V.M. Geneticheskie algoritmy / Pod red. V.M. Kurejchika. – 2-e izd., ispr. i dop. – M.: Fizmatlit, 2006. – 320 ps.
2. Geneticheskij algoritm. Standart: Ch. 1: Opisanie standartnogo geneticheskogo algoritma (sGA) / A. B. Sergienko, P. V. Galushin, V. V. Buhtojarov; Sibirskij gos. ajerokosmicheskij un-t im. M. F. Reshetneva. – Krasnojarsk, 2010. - 384 ps.
3. Rutkovsky D., Pilinsky M., Rutkovsky L. Neural networks, genetic algorithms and unsharp systems: The lane with polsk. I.D.Rudinskogo. — M.: A hot line — the Telecom, 2006. — 452 ps.
4. Gray Code – Wikipedia, Available at: [https://en.wikipedia.org/wiki/Gray\\_code](https://en.wikipedia.org/wiki/Gray_code) (accessed 4 April 2021)
5. M. Wall. Galib: A C++ library of genetic algorithm components. Mechanical Engineering Department, Massachusetts Institute of Technology, 87, 1996.
6. Whitley, D (1999): A free lunch proof for Gray versus binary encodings, Proc. Genetic and Evolutionary Computation (GECCO-1999), pp. 726-733.
7. Gvozdzak P. Goddyn L. Binary gray codes with long bit runs. Electr. J. Comb, 10:1–10, 2003.
8. Rana, S. and Whitley, D. (1997). Bit Representations with a Twist. In Ba ĉk, T., editor, Proceedings of the Seventh International Conference on Genetic Algorithms, pages 188– 195. Morgan Kaufmann.