



Выпускная квалификационная работа

Применение методов оптимизации для формирования обобщенных кодов Баркера

Студент: Кирилин Глеб Андреевич

Группа КМБО-01-17

Научный руководитель: Сенявин М.М.

Актуальность задачи

Актуальность предлагаемого исследования обусловлена широким использованием последовательностей Баркера в радиолокации, цифровой речи, ультразвуковом сканировании, GPS, Wi-Fi.

Коды Баркера длиной N , равной 11 и 13, используются в радиолокационных системах с расширенным спектром прямой последовательности.

Цель работы

Целью работы является разработка алгоритмов и программ, реализующих методы оптимизации, применимых к задаче об обобщении кодов Баркера.

Последовательности Баркера

Последовательность Баркера – это числовая последовательность x_1, x_2, \dots, x_N , где каждый элемент равен 1 или -1, причем

$$|ACF_x(\mathbf{k})| \leq 1$$

где

$$ACF_x(\mathbf{k}) = \sum_{i=1}^{N-|k|} x_i x_{i+|k|}$$

для $1 - N \leq k \leq N - 1$, кроме 0.

Известные последовательности Баркера

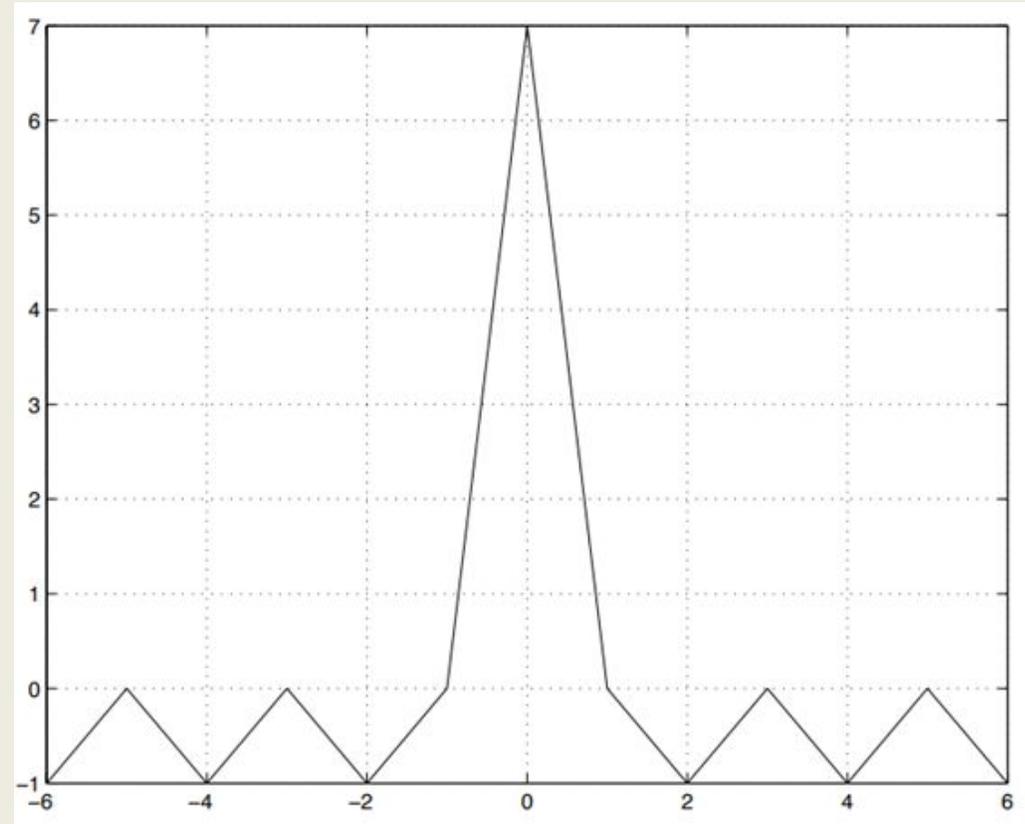
С точностью до реверсирования порядка и смены знаков каждого из элементов на данный момент известно только 9 кодов Баркера.

Длина	Последовательности	
2	+1 -1	+1 +1
3	+1 +1 -1	
4	+1 -1 +1 +1	+1 -1 -1 -1
5	+1 +1 +1 -1 +1	
7	+1 +1 +1 -1 -1 +1 -1	
11	+1 +1 +1 -1 -1 -1 +1 -1 -1 +1 -1	
13	+1 +1 +1 +1 +1 -1 -1 +1 +1 -1 +1 -1 +1	

Известные последовательности Баркера

График автокорреляционной функции для последовательности Баркера длины 7.

$$ACF_x(\mathbf{0}) = N'$$



Постановка основной задачи

$$x = (x_1, x_2, \dots, x_n), x \in E^n$$

$$f_j(x) = \sum_{i=1}^{n-j} x_i x_{i+j}, 1 \leq j \leq n-1$$

$$x_i^2 - 1 = 0, \quad 1 \leq i \leq n$$

Требуется найти вектор x , при котором функция $F(x)$ принимает наименьшее возможное значение среди всех тех значений, которые она принимает для допустимых x :

$$F(x) = \max_j |f_j(x)| \rightarrow \min$$

Решение основной задачи

Алгоритм:

При решении были использованы метод полного перебора и метод покоординатного спуска – замена x_k на $-x_k$.

При росте N время, затраченное на поиск нужной последовательности методом полного перебора, становится большим.

Поэтому перейдем к алгоритму методу покоординатного спуска.

Решение основной задачи

Суть метода поординатного спуска заключается в том, чтобы заменять координату x_k на $-x_k$ и идти в сторону уменьшения критерия качества.

Если

критерий качества уменьшился, то координату $-x_k$ оставляем и

переходим к

x_{k-1} , иначе возвращаем начальное значение x_k и переходим к x_{k-1} .

Решение основной задачи

Вычисления:

N = 7:

Для начальной точки $x = (1, 1, 1, 1, 1, 1, 1)$ получилась тупиковая точка $\dot{x} = (1, 1, 1, -1, -1, 1, -1)$, минимум $F(\dot{x}) = 1$
Точка \dot{x} также является кодом Баркера

N = 11:

Для начальной точки $x = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$ получилась тупиковая точка $\dot{x} = (1, 1, 1, -1, 1, 1, 1, -1, -1, 1, -1)$, минимум $F(\dot{x}) = 3$

N = 19:

Для начальной точки $x = (1, 1, 1, \dots, 1, -1, -1)$ получилась тупиковая точка $\dot{x} = (1, 1, 1, 1, -1, 1, 1, -1, 1, -1, -1, 1, 1, 1, -1, -1, 1, -1, -1)$, минимум $F(\dot{x}) = 3$

Вывод

- Преимущество направленного перебора перед полным – при программной реализации алгоритма смогли избежать полного перебора всех функций.
- Не все тупиковые точки являются оптимальными.
- Минимум функции $F(x)$ сильно зависит от выбора начальной точки.

Полученные тупиковые точки можно использовать как начальные приближения в более серьезных методах оптимизации.

Перейдем к эквивалентной задаче, которая имеет стандартный вид и гладкий критерий качества.

Постановка эквивалентной задачи

$$\begin{aligned} x_{n+1} &\rightarrow \min \\ f_j^2(x) - x_{n+1}^2 &\leq 0, 1 \leq j \leq n-1 \\ x_i^2 - 1 &= 0, 1 \leq i \leq n \end{aligned}$$

Метод штрафных функций

$$g_j(x) = \left(\max \left((f_j^2(x) - x_{n+1}^2), 0 \right) \right)^2, 1 \leq j \leq n-1$$

$$g_j(x) = (x_{j+1-n}^2 - 1)^2, n \leq j \leq 2n-1$$

$$G_k(x) = x_{n+1} + k \sum_{j=1}^{2n-1} g_j(x)$$

$$G_k(x) \rightarrow \min, k \rightarrow \infty$$

Описание метода штрафных функций

Имеется задача

$$f(x) \rightarrow \min$$

При ограничениях

$$\begin{aligned} f_i(x) &\leq 0, 1 \leq i \leq n \\ f_i(x) &= 0, n + 1 \leq i \leq m \end{aligned}$$

Тогда целесообразно использовать штрафную функцию такого вида:

$$\alpha(x) = \sum_{i=1}^n g_i(x) + \sum_{i=n+1}^m g_i(x) = \sum_{i=1}^n (\max(f_i(x), 0))^p + \sum_{i=n+1}^m |f_i(x)|^p$$

Далее переходим от исходной задачи к задаче безусловной

минимизации $G_k(x) = f(x) + k\alpha(x) \rightarrow \min$, где $0 < k \rightarrow \infty$

Алгоритм метода штрафных функций

Начальные данные

Выбираем $\epsilon > 0$, начальную точку x_0 и параметр штрафа.

Первый шаг

Решаем задачу

$$G_k(x) = f(x) + k\alpha(x) \rightarrow \min$$

Полагаем, что новый x равен оптимальному решению задачи и переходим к следующему шагу.

Второй шаг

Если $k\alpha(x) < \epsilon$, то останавливаемся.

В противном случае увеличиваем k и переходим к первому шагу.

Решение эквивалентной задачи

Алгоритм нахождения вектора для основной задачи :

1. Взять несколько тупиковых точек, которые были получены при решении основной задачи.
2. Взять эти точки в качестве начальных для метода штрафных функций.
3. Составить штрафную функцию и минимизировать ее.
4. Округлить полученные значения компонент вектора до 1 или -1.
5. Продолжить применение метода штрафных функций с полученной точкой в качестве начальной, если результат не улучшился – значит, получен квазиоптимум.

Решение эквивалентной задачи

Вычисления:

N = 11

Начальный вектор: 1, 1, 1, -1, 1, -1, -1, 1, -1, -1, -1

Минимум = 3

Итоговый вектор: 1, 1, 1, -1, 1, -1, -1, 1, -1, -1, -1

Минимум = 3

N = 19

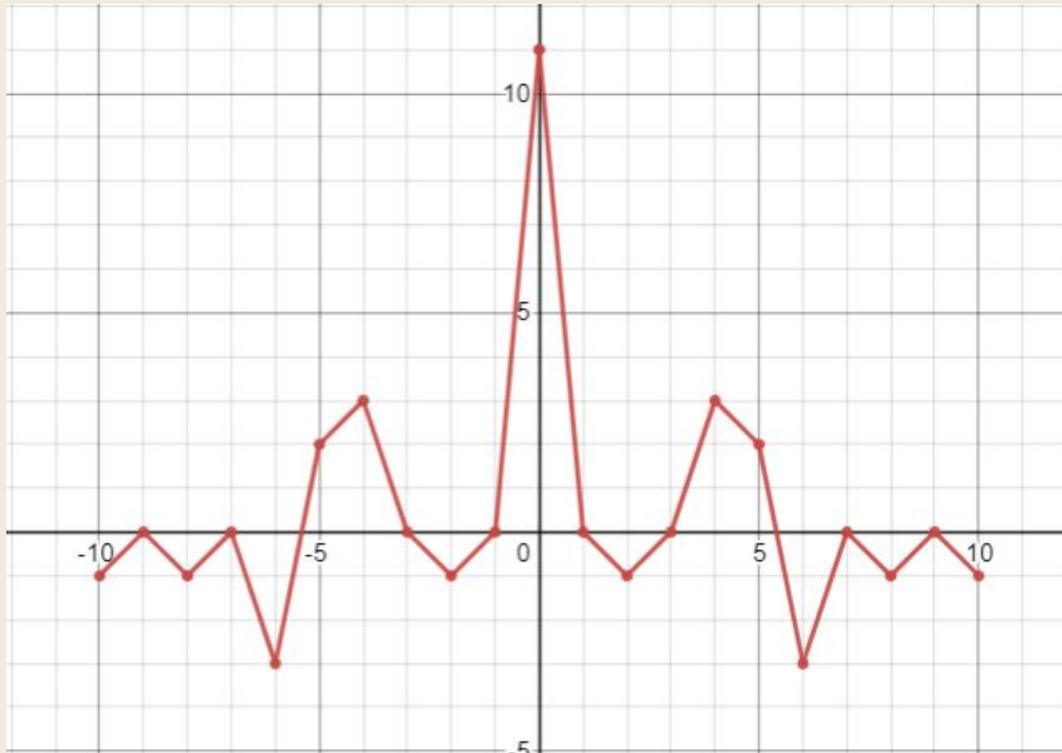
Начальный вектор: 1, 1, 1, 1, -1, 1, 1, -1, 1, -1, -1, 1, 1, 1, -1, -1, 1, -1, -1

Минимум = 4

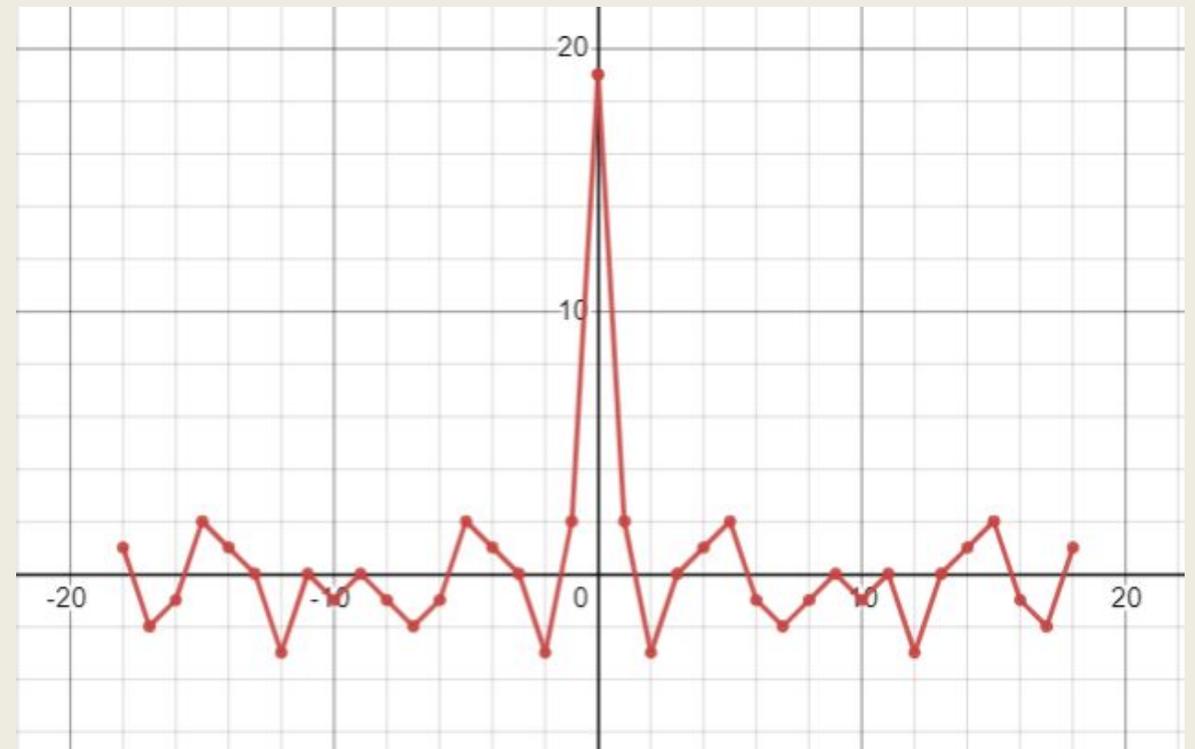
Итоговый вектор: -1, 1, 1, 1, 1, 1, 1, -1, 1, -1, -1, 1, 1, -1, -1, -1, 1, 1, -1

Минимум = 3

Решение эквивалентной задачи



ACF для последовательности длиной $N = 11$



ACF для последовательности длиной $N = 19$

Решение эквивалентной задачи

N = 30

Начальный вектор: 1, 1, 1, 1, 1, 1, -1, 1, 1, -1, -1, 1, 1, 1, 1, -1, -1, -1, 1, -1, 1, -1,
-1, 1, 1, 1, -1, -1, 1, -1. *Минимум = 6*

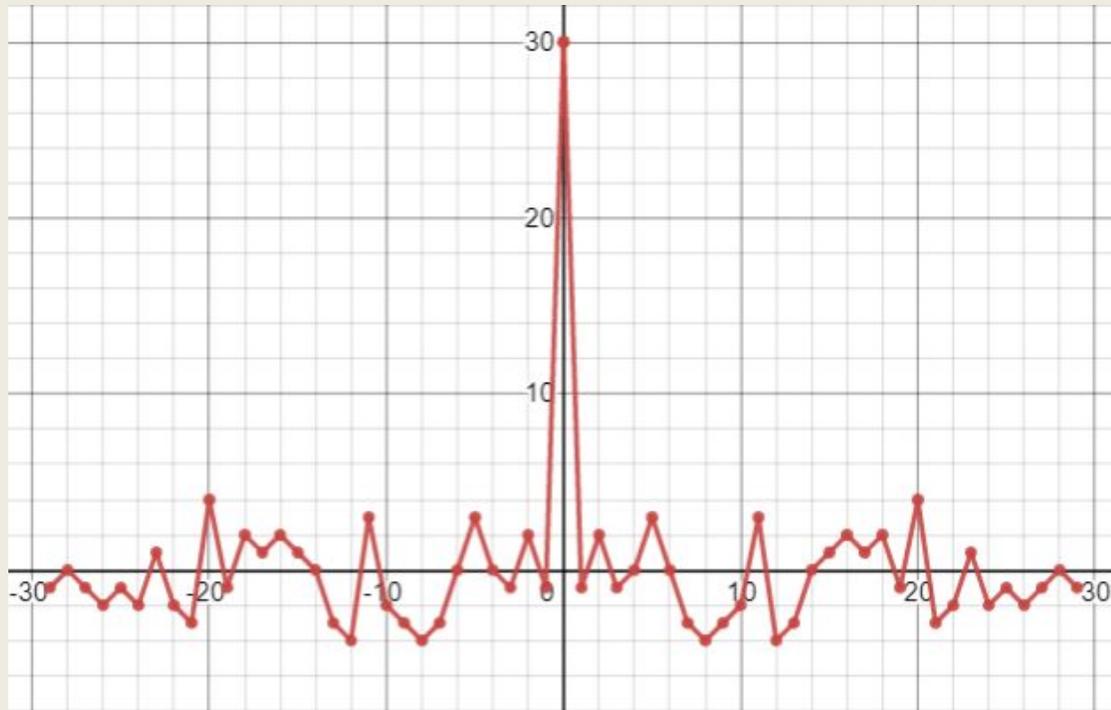
Итоговый вектор: 1, 1, 1, 1, 1, 1, -1, -1, 1, -1, -1, -1, 1, -1, 1, -1, -1, -1, 1, 1, 1, -1,
-1, 1, -1, 1, -1, -1, 1, -1. *Минимум = 4*

N = 40

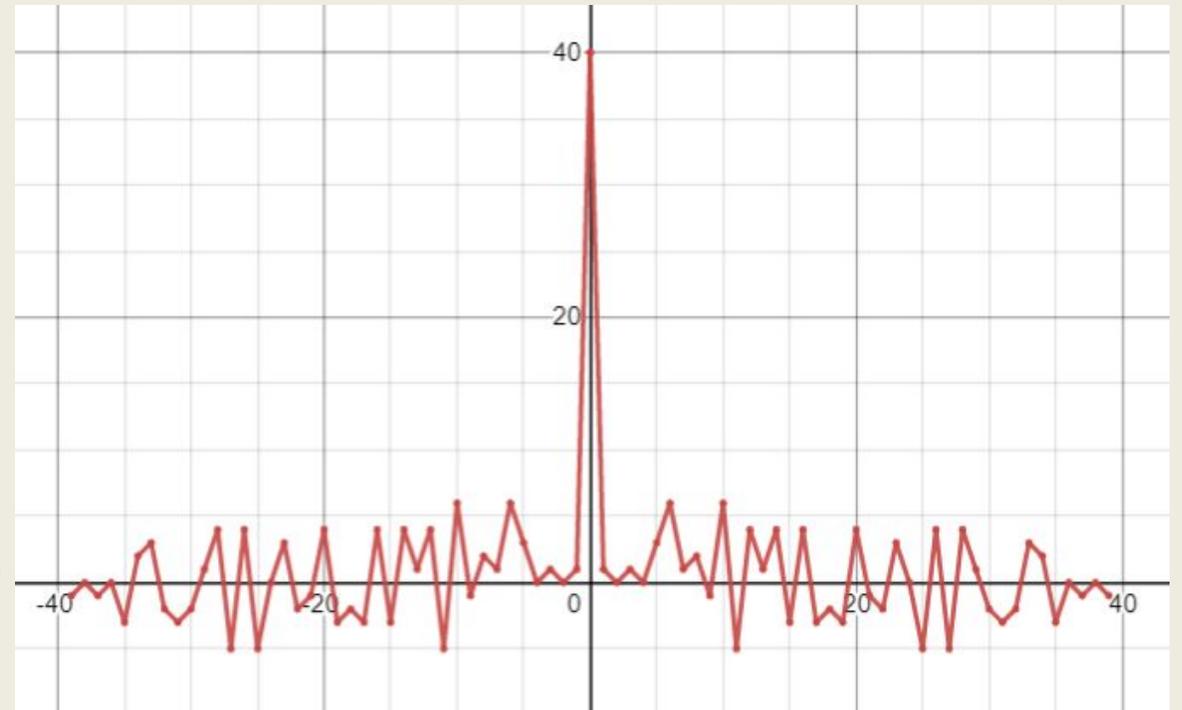
Начальный вектор: 1, 1, 1, -1, 1, 1, 1, 1, 1, 1, 1, -1, 1, -1, 1, 1, -1, -1, 1, -1, -1, 1, 1, 1, 1,
-1, -1, -1, 1, -1, 1, -1, -1, 1, 1, 1, -1, -1, 1, -1. *Минимум = 6*

Итоговый вектор: 1, 1, 1, -1, 1, 1, 1, 1, 1, 1, 1, -1, 1, -1, 1, 1, -1, -1, 1, -1, -1, 1, 1, 1, 1, -1,
-1, -1, 1, -1, 1, -1, -1, 1, 1, 1, -1, -1, 1, -1. *Минимум = 6*

Решение эквивалентной задачи



ACF для кода Баркера длиной $N = 30$



ACF для кода Баркера длиной $N = 40$

Решение эквивалентной задачи

N = 45

Начальный вектор: 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, -1, -1, 1, -1, -1, 1, 1, 1, -1, -1, -1, 1, -1, 1, -1, -1, 1, 1, 1, -1, -1, 1, -1. *Минимум = 6*

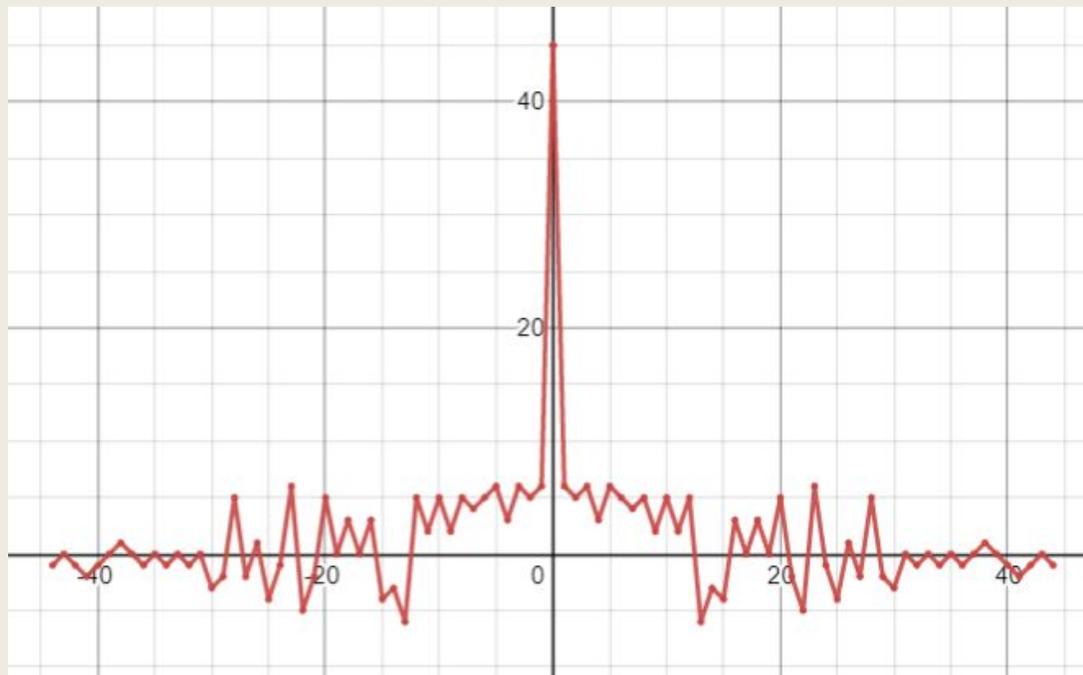
Итоговый вектор: 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, -1, -1, 1, -1, -1, 1, 1, 1, 1, -1, -1, -1, 1, -1, 1, -1, -1, 1, 1, 1, -1, -1, 1, -1. *Минимум = 6*

N = 50

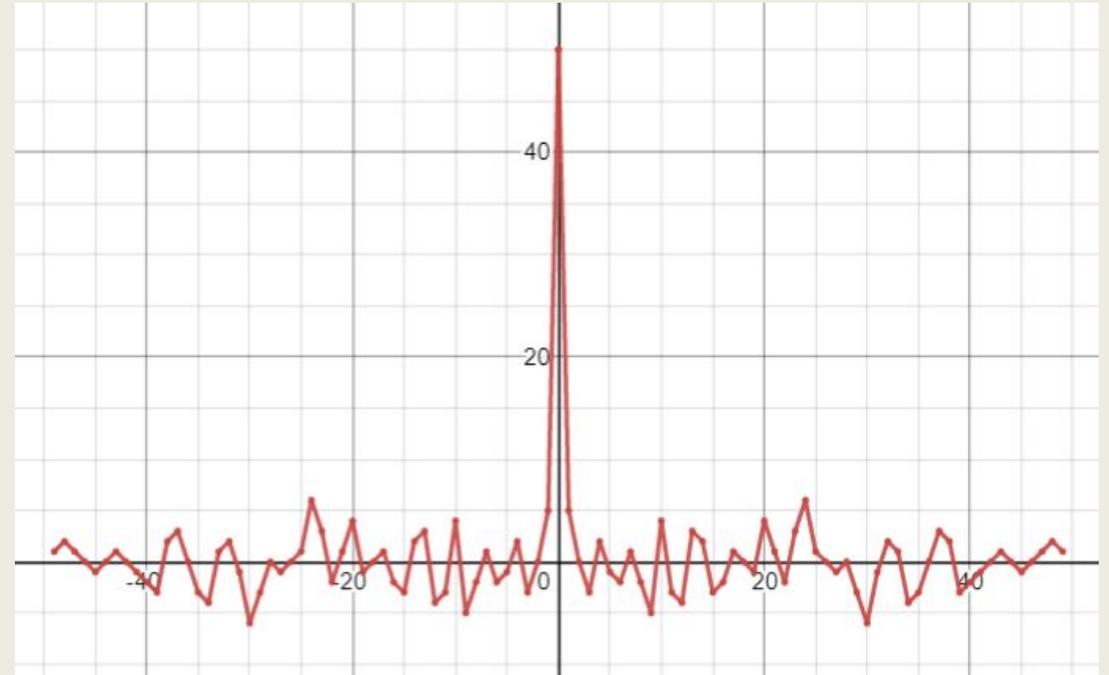
Начальный вектор: 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, 1, 1, 1, -1, 1, -1, -1, 1, -1, 1, -1, 1, 1, -1, -1, 1, -1, -1, 1, 1, 1, 1, -1, -1, -1, 1, -1, 1, -1, -1, -1, 1, -1. *Минимум = 8*

Итоговый вектор: 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, 1, -1, -1, -1, 1, -1, 1, 1, -1, 1, -1, 1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, 1, 1, -1, -1, -1, 1, 1. *Минимум = 6*

Решение эквивалентной задачи



ACF для кода Баркера длиной $N = 45$



ACF для кода Баркера длиной $N = 50$

Выводы

- Применены современные методы оптимизации к эквивалентной задаче.
- Рассмотрены случаи для $N > 19$.
- Для случая $N = 11$ критерий качества не был улучшен.
- Для $N = 19, 30, 50, 60$ применение метода штрафных функций позволило улучшить критерий качества и приблизиться к минимуму, но все же его не достигнуть.
- Для $N = 40, 45$ критерий качества либо не менялся, либо только ухудшался.

Спасибо за внимание!