

# **Лабораторная работа №1**

## **Простой генетический алгоритм**

### **Общие сведения**

ГА используют принципы и терминологию, заимствованные у биологической науки – генетики. В ГА каждая особь представляет потенциальное решение некоторой проблемы. В классическом ГА особь кодируется строкой двоичных символов – хромосомой, каждый бит которой называется геном. Множество особей – потенциальных решений составляет популяцию. Поиск (суб)оптимального решения проблемы выполняется в процессе эволюции популяции - последовательного преобразования одного конечного множества решений в другое с помощью генетических операторов репродукции, кроссинговера и мутации.

ЭВ используют следующие механизмы естественной эволюции:

1) Первый принцип основан на концепции выживания сильнейших и естественного отбора по Дарвину, который был сформулирован им в 1859 году в книге «Происхождение видов путем естественного отбора». Согласно Дарвину особи, которые лучше способны решать задачи в своей среде, выживают и больше размножаются (репродуцируют). В генетических алгоритмах каждая особь представляет собой решение некоторой проблемы. По аналогии с этим принципом особи с лучшими значениями целевой (фитнесс) функции имеют большие шансы выжить и репродуцировать. Формализация этого принципа, как мы увидим далее, дает оператор репродукции.

2) Второй принцип обусловлен тем фактом, что хромосома потомка состоит из частей полученных из хромосом родителей. Этот принцип был открыт в 1865 году Менделем. Его формализация дает основу для оператора скрещивания (кроссинговера).

3) Третий принцип основан на концепции мутации, открытой в 1900 году де Вре. Первоначально этот термин использовался для описания существенных (резких) изменений свойств потомков и приобретение ими свойств, отсутствующих у родителей. По аналогии с этим принципом генетические алгоритмы используют подобный механизм для резкого изменения свойств потомков и тем самым, повышают разнообразие (изменчивость) особей в популяции (множестве решений).

Эти три принципа составляют ядро ЭВ. Используя их, популяция (множество решений данной проблемы) эволюционирует от поколения к поколению.

Эволюцию искусственной популяции – поиска множества решений некоторой проблемы формально можно описать алгоритмом, который представлен на рис.1.1.

ГА берет множество параметров оптимизационной проблемы и кодирует их последовательностями конечной длины в некотором конечном алфавите (в простейшем случае двоичный алфавит «0» и «1»).

Предварительно простой ГА случайным образом генерирует начальную популяцию стрингов (хромосом). Затем алгоритм генерирует следующее поколение (популяцию), с помощью трех основных генетических операторов:

- 1) Оператор репродукции (ОР);
- 2) Оператор скрещивания (кроссинговера, ОК);
- 3) Оператор мутации (ОМ).

Генетические операторы являются математической формализацией приведенных выше трех основополагающих принципов Дарвина, Менделя и де Вре естественной эволюции.

ГА работает до тех пор, пока не будет выполнено заданное количество поколений (итераций) процесса эволюции или на некоторой генерации будет получено заданное качество или вследствие преждевременной сходимости при попадании в некоторый локальный оптимум.

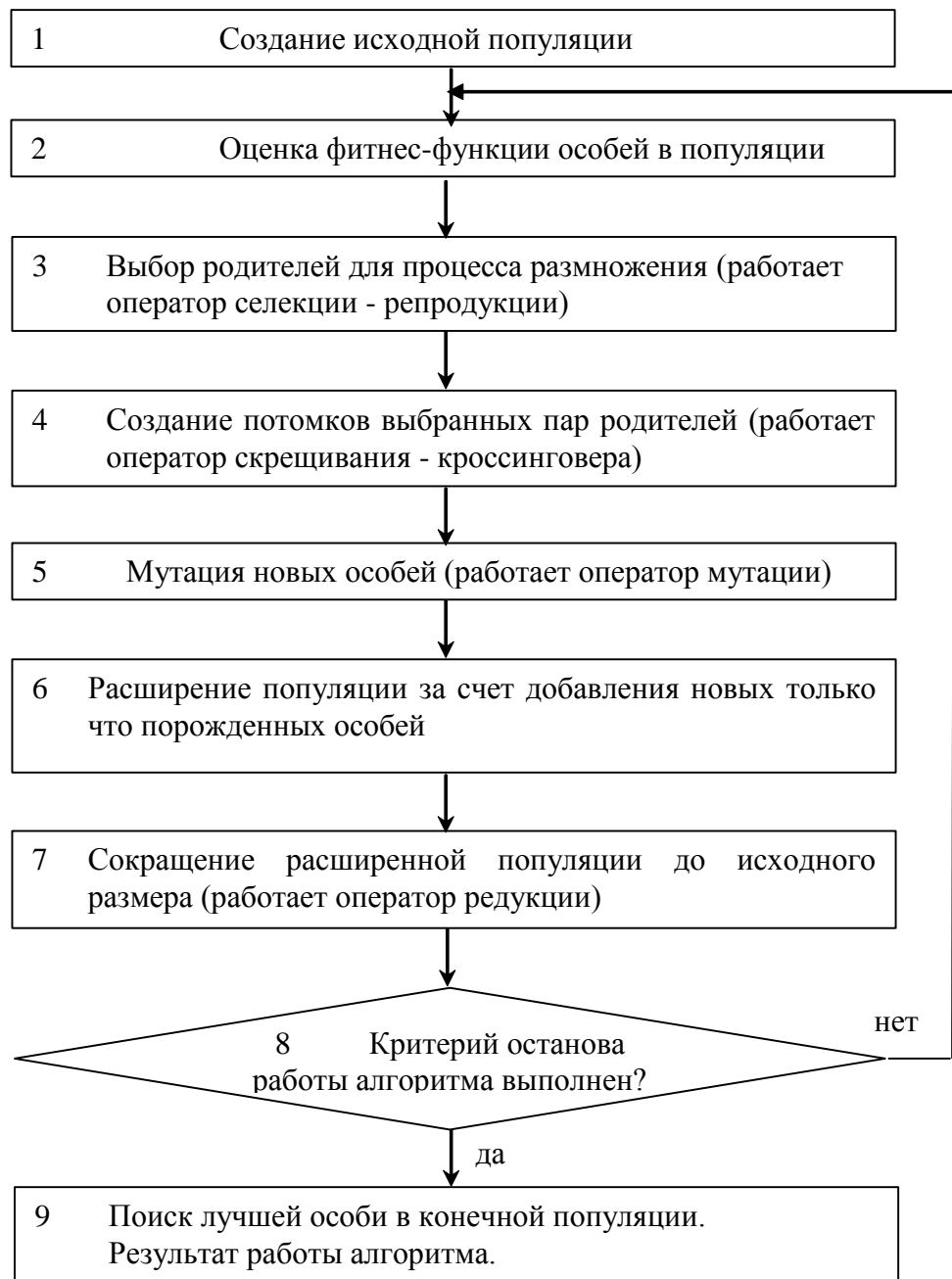


Рис.1.1. Простой генетический алгоритм

В каждом поколении множество искусственных особей создается с использованием старых и добавлением новых с хорошими свойствами. Генетические алгоритмы - не просто случайный поиск, они эффективно используют информацию накопленную в процессе эволюции.

В отличие от других методов оптимизации ГА оптимизируют различные области пространства решений одновременно и более

приспособлены к нахождению новых областей с лучшими значениями целевой функции за счет объединения квазиоптимальных решений из разных популяций.

## **Основная терминология в генетических алгоритмах**

Поскольку, как отмечалось выше, в ГА используются некоторые термины, заимствованные из генетики, то полезно привести их дать техническую интерпретацию.

**Ген** – элементарный код в хромосоме  $s_i$ , называемый также знаком, или детектором (в классическом ГА  $s_i=0,1$ ).

**Хромосома** – упорядоченная последовательность генов в виде закодированной структура данных  $S=(s_1, s_2, \dots, s_n)$ , определяющая решение (в простейшем случае двоичная последовательность –стринг, где  $s_i=0,1$ ).

**Локус** – местоположение (позиция, номер бита) данного гена в хромосоме.

**Аллель** – значение, которое принимает данный ген (например,  $\{0,1\}$ ).

**Особь** – одно потенциальное решение проблемы (представляемое хромосомой).

**Популяция** – множество особей – потенциальных решений, которые представляются хромосомами.

**Поклоение** – текущая популяция ГА (для текущей итерации алгоритма).

**Генотип** – набор хромосом данной особи (особями популяции могут быть генотипы либо отдельные хромосомы).

**Генофонд** – множество всех возможных генотипов.

**Фенотип** – набор значений, соответствующий данному генотипу, - декодированное множество параметров или структура данной задачи (например, фенотип - десятичное значение  $x$  его двоичного кода - геннотипа);

**Размер (мощность)** популяции  $N$  – число особей (решений) в популяции.

**Число поколений (генераций)** – количество итераций, в течение которых производится генетический поиск.

**Селекция** – комплекс правил, моделирующих выживание особей на основе их значений ЦФ.

**Эволюция популяции** – это чередование поколений, в которых хромосомы изменяют свои признаки, чтобы каждая новая популяция наилучшим образом приспособлялась к внешней среде.

**Фитнесс-функция** (полезности) – важнейшее понятие, определяющее меру приспособленности данной особи в популяции. В задачах оптимизации часто представляется целевой функцией или определяет меру близости к оптимальному решению. В обучении может принимать вид функции погрешности (ошибки). На каждой итерации ГА приспособленность каждой особи популяции оценивается с помощью фитнес-функции.

## Генетические операторы

Рассмотрим работу простого ГА на следующем примере. Надо найти (для простоты) целочисленное значение  $x$  в интервале от 0 до 31, при котором функции  $y = x^2$  принимает максимальное значение.

Здесь особи начальной популяции (двоичные коды значений переменных  $x$  - столбец 2) сгенерированы случайным образом. Двоичный код значения  $x$  называется хромосомой (она представляет генотип). Популяция образует множество потенциальных решений данной проблемы. В третьем столбце представлены их десятичные значения (фенотип). Далее на этом примере проиллюстрируем работу трех основных генетических операторов.

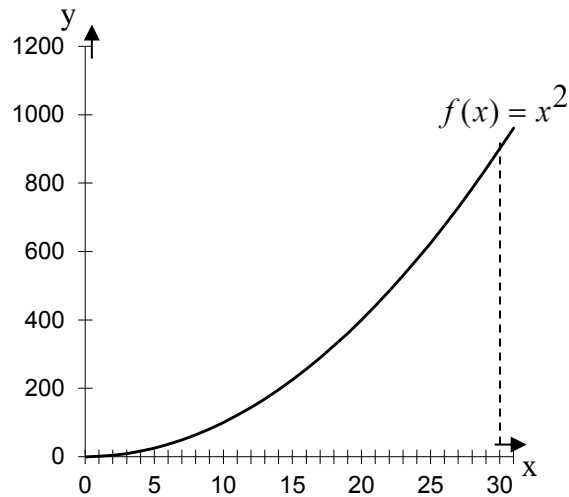


Рис.1.2. Пример функции

Начальный этап работы ГА для данного примера приведен в верхней таблице (репродукция) рис.1.3.

### Репродукция

Репродукция – это процесс, в котором хромосомы копируются в промежуточную популяцию для дальнейшего "размножения" согласно их значениям целевой (фитнес-) функции. При этом хромосомы с лучшими значениями целевой функции имеют большую вероятность попадания одного или более потомков в следующее поколение.

### Репродукция

№ хромо- сомы	Начальная популяция особей	Десятичное значение x	Значение f(x)	$\frac{f(x_i)}{\sum f(x_j)}$	Среднее значение $\overline{f(x)}$	Максимально е значение $f_{\max}(x)$
1	01101	13	169	0,14	293	576
2	11000	24	576	0,49		
3	01000	8	64	0,06		
4	10011	19	361	0,31		

### Кроссинговер

№ хромо- сомы	Популяция после репродукции	Десятичное значение x	Значение f(x)	Пары хромосом для кроссинговера	Среднее значение $\overline{f(x)}$	Максимальное значение $f_{\max}(x)$
1	0 1 1 0 1	13	169	1-2	439	729
2	1 1 0 0 0	24	576	1-2		
3	1 1 0 0 0	24	576	3-4		
4	1 0 0 1 1	19	361	3-4		

### Мутация

№ хромо- сомы	Популяция после кроссинговера	Новая популяция после мутации	Десятичное значение x	Значение f(x)	Среднее значение $\overline{f(x)}$	Максимальное значение $f_{\max}(x)$
1	01100	01100	12	144	496.5	961
2	11001	11001	25	625		
3	11011	11111	31	961		
4	10000	10000	16	256		

Рис.1.3. Эволюция популяции

Очевидно, оператор репродукции (ОР) является искусственной версией естественной селекции – выживания сильнейших по Дарвину. Этот оператор представляется в алгоритмической форме различными способами. Самый

простой (и популярный) метод реализации ОР – построение колеса рулетки, в которой каждая хромосома имеет сектор, пропорциональный ее значению ЦФ. Для нашего примера "колесо рулетки" имеет следующий вид, представленный на рис.1.4.

Для селекции хромосом используется случайный поиск на основе колеса рулетки. При этом колесо рулетки вращается и после останова ее указатель определяет хромосому для селекции в промежуточную популяцию (родительский пул). Очевидно, что хромосома, которой соответствует больший сектор рулетки, имеет большую вероятность попасть в следующее поколение. В результате выполнения оператора репродукции формируется промежуточная популяция, хромосомы которой будут использованы для построения поколения с помощью операторов скрещивания.

В нашем примере выбираем хромосомы для промежуточной популяции, вращая колесо рулетки 4 раза, что соответствует мощности начальной популяции. Величину  $\frac{f(x_i)}{\sum f(x_j)}$  обозначим, как  $P(x_i)$ , тогда ожидаемое количество копий  $i$ -ой хромосомы определяется  $M=P(x_i)*N$ ,  $N$ -мощность популяции. Число копий хромосомы, переходящих в следующее поколение, иногда определяется и так:

$$\tilde{M} = \frac{f(x_i)}{\bar{f}(x)},$$

где  $\bar{f}(x)$ - среднее значение хромосомы в популяции.

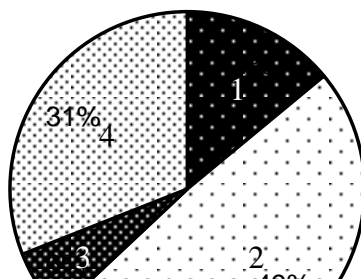




Рис.1.4. Колесо рулетки

Расчетные числа копий хромосом по приведенной формуле следующие: хромосома 1 - 0,56; хромосома 2 - 1,97; хромосома 3 - 0,22; хромосома 4 - 1,23. В результате, в промежуточную популяцию 1-я хромосома попадает в одном экземпляре, 2-я - в двух, 3-я - совсем не попадает, 4-я - в одном экземпляре. Полученная промежуточная популяция является исходной для дальнейшего выполнения операторов кроссинговера и мутации.

### Оператор кроссинговера (скрещивания)

Одноточечный или простой оператор кроссинговера (ОК) с заданной вероятностью  $P_c$  выполняется в 3 этапа:

1-й этап. Две хромосомы (родители)

$$\begin{array}{l} A = a_1 a_2 \dots a_k \quad a_{k+1} \dots a_L \\ B = b_1 b_2 \dots b_k \quad b_{k+1} \dots b_L \end{array}$$

Точка кроссинговера

выбираются случайно из промежуточной популяции, сформированной при помощи оператора репродукции (ОР).

2-й этап. Случайно выбирается точка скрещивания - число  $k$  из диапазона  $[1, 2 \dots n-1]$ , где  $n$  - длина хромосомы (число бит в двоичном коде)

3-й этап. Две новых хромосомы  $A'$ ,  $B'$  (потомки) формируются из  $A$  и  $B$  путем обмена подстрок после точки скрещивания:

$$A' = a_1 a_2 \dots a_k b_{k+1} \dots b_L$$

$$B' = b_1 b_2 \dots b_k a_{k+1} \dots a_L$$

Например, рассмотрим выполнение кроссинговера для хромосом 1 и 2 из промежуточной популяции:

$$A = 0 \ 1 \ 1 \ 0 \ 1$$

$$B = 1 \ 1 \ 0 \ 0 \ 0$$

$$1 \leq k \leq 4, \quad k=4$$

$$A' = 0 \ 1 \ 1 \ 0 \ 0$$

$$B' = 1 \ 1 \ 0 \ 0 \ 1$$

Следует отметить, что ОК выполняется с заданной вероятностью  $P_c$  (отобранные два родителя не обязательно производят потомков). Обычно величина  $P_c \approx 0.5$ .

Таким образом, операторы репродукции и скрещивания очень просты – они выполняют копирование особей и частичный обмен частей хромосом. Продолжение нашего примера представлено на рис.1.3 во второй таблице (кроссинговер).

Сравнение с предыдущей таблицей показывает, что в промежуточной популяции после скрещивания улучшились все показатели популяции (среднее и максимальное значения ЦФ).

### Мутация

Далее согласно схеме классического ГА с заданной вероятностью  $P_m$  выполняется оператор мутации. Иногда этот оператор играет вторичную роль. Обычно вероятность мутации мала -  $P_m \approx 0,001$ .

Оператор мутации (ОМ) выполняется в 2 этапа:

1-й этап. В хромосоме  $A = a_1 a_2 \dots a_L$  случайно выбирается  $k$ -ая позиция (бит) мутации ( $1 \leq k \leq n$ ).

2-й этап. Производится инверсия значения гена в k-й позиции.

$$a'_k = \bar{a}_k.$$

Например, для хромосомы 11011 выбирается  $k=3$  и после инверсии значения третьего бита получается новая хромосома – 11111. Продолжение нашего примера представлено в третьей таблице (мутация) рис.1.3. Таким образом, в результате применения генетических операторов найдено оптимальное решение  $x=31$ .

В данном случае, поскольку пример искусственно подобран, мы нашли оптимальное решение за одну итерацию. В общем случае ГА работает до тех пор, пока не будет достигнут критерий окончания процесса поиска и в последнем поколении определяется лучшая особь.

### **Представление вещественных решений в двоичной форме**

В предыдущем примере мы рассматривали только целочисленные решения. Обобщим ГА на случай вещественных чисел на следующем примере, в котором для функции

$$f(x)=(1,85-x)*\cos(3,5x-0,5),$$

представленной на рис.1.5 необходимо найти вещественное  $x \in [-10,+10]$ ,

которое максимизирует  $f$ , т.е. такое  $x_0$ , для которого  $f(x_0) \geq f(x)$  для всех  $x \in [-10,+10]$ .

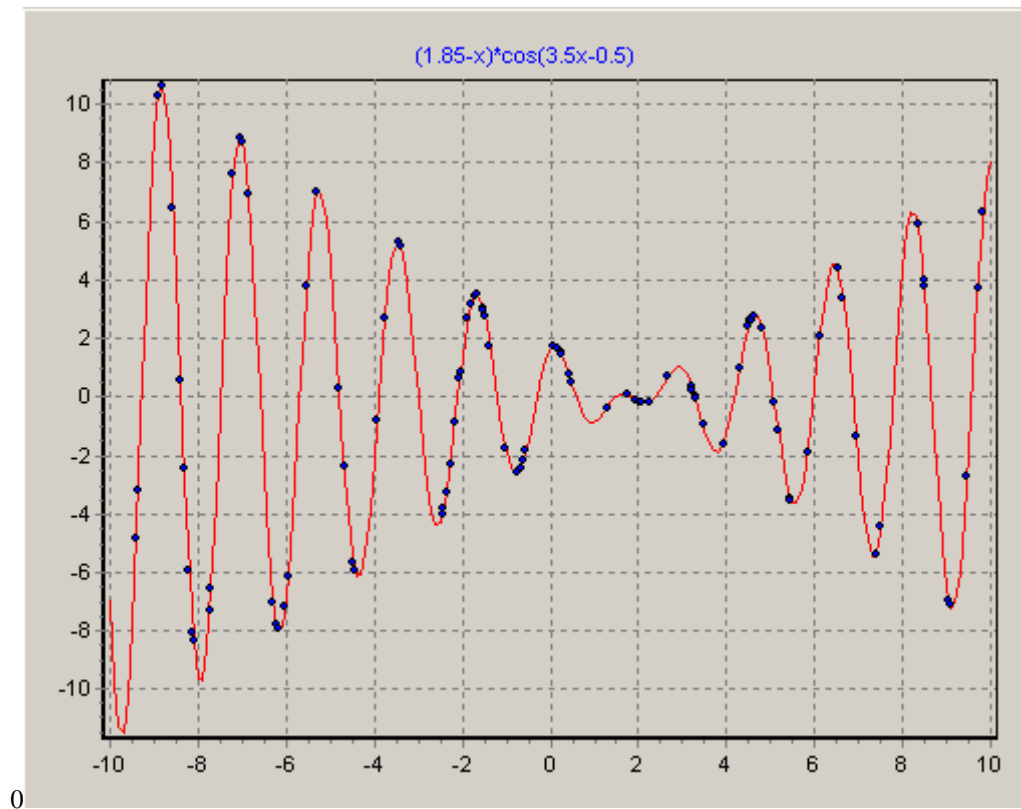


Рис.1.5. Пример функции с популяцией особей в начале эволюции

Нам необходимо построить ГА для решения этой задачи. Для представления вещественного решения (хромосомы)  $x$  будем использовать двоичный вектор, который применяется в классическом простом ГА. Его длина зависит от требуемой точности решения, которую в данном случае положим 3 знака после запятой.

Поскольку отрезок области решения имеет длину 20, для достижения заданной точности отрезок  $[-10, +10]$  должен быть разбит на равные части, число которых должно быть не менее  $20 \cdot 1000$ . В качестве двоичного представления используем двоичный код номера отрезка. Этот код позволяет определить соответствующее ему вещественное число, если известны границы области решения. Отсюда следует, что двоичный вектор для кодирования вещественного решения должен иметь 15 бит, поскольку

$$16384 = 2^{14} < 20000 \leq 2^{15} = 32768$$

Это позволяет разбить отрезок  $[-10, +10]$  на 32768 частей и обеспечить необходимую точность. Отображение из двоичного представления  $(b_{14}b_{13}...b_0)$  (

$b_i \in \{0,1\}$ ) в вещественное число из отрезка  $[-10,+10]$  выполняется в два шага.

1) Перевод двоичного числа в десятичное:

$$(<b_{14}b_{13}..b_0>)_2 = \left( \sum_{i=0}^{14} b_i 2^i \right)_{10} = X'$$

2) Вычисление соответствующего вещественного числа  $x$ :

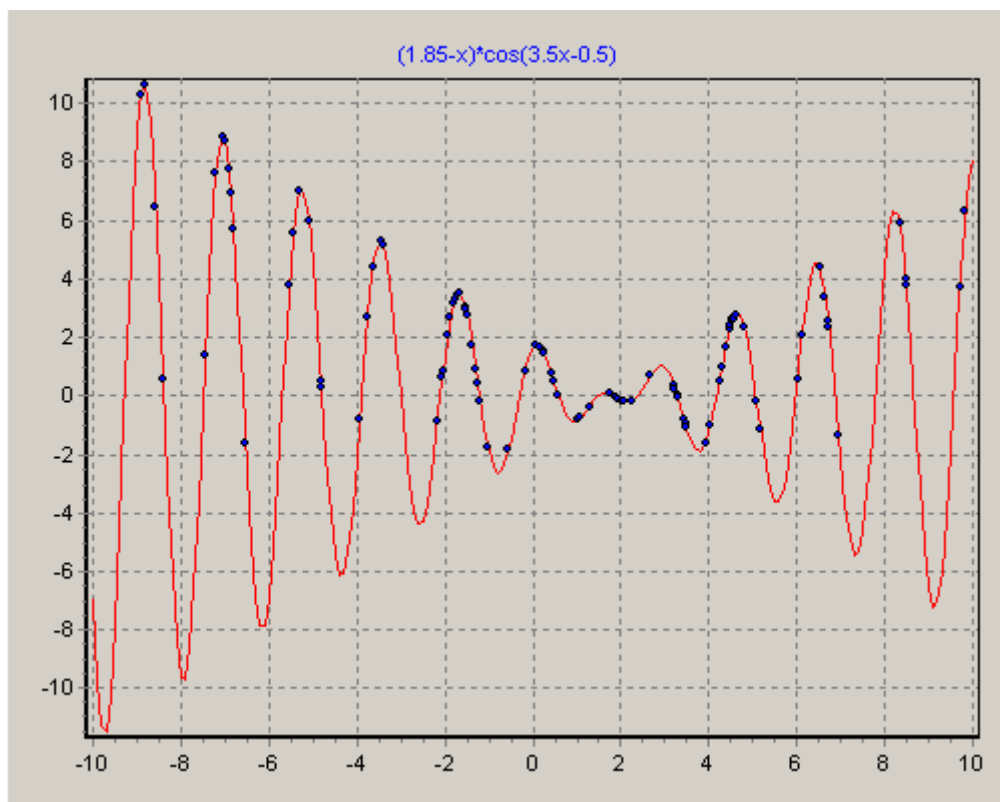
$$x = -10 + x' \cdot \frac{10 - (-10)}{2^{15} - 1}, \text{ где } -10 \text{ левая граница области решения.}$$

Естественно хромосомы

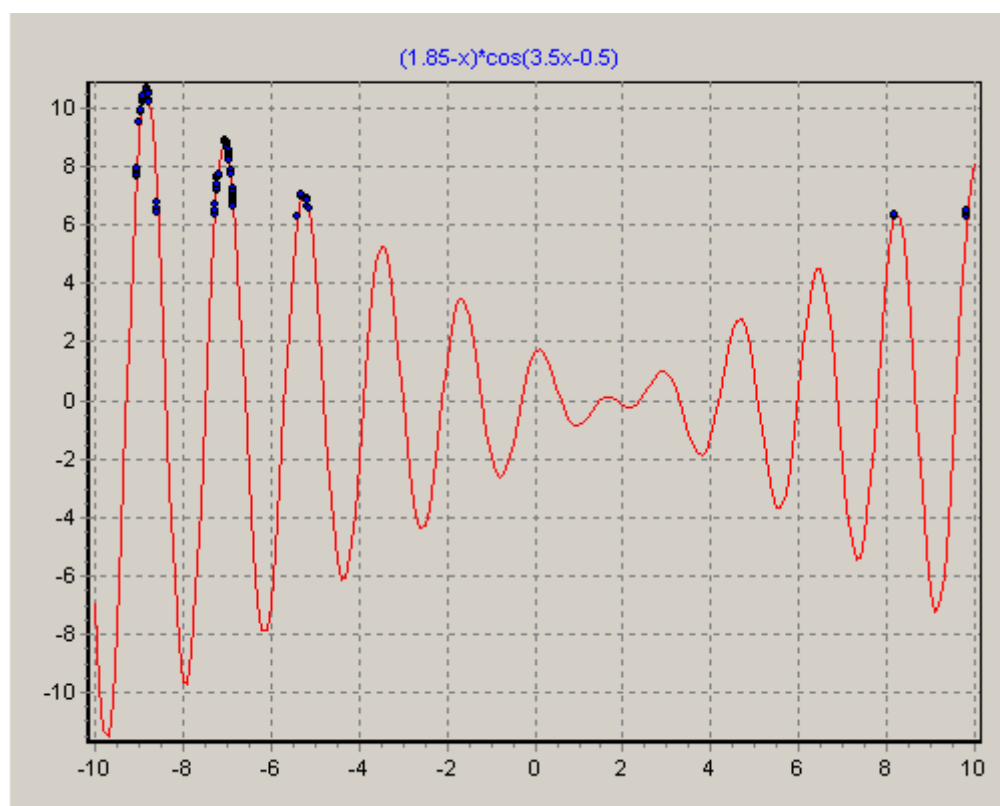
(0000000000000000) и (1111111111111111)

представляют границы отрезка  $-10$  и  $+10$  соответственно.

Очевидно, при данном двоичном представлении вещественных чисел можно использовать классический простой ГА. На рис.1.5–рис.1.8 представлено расположение особей - потенциальных решений на различных этапах ГА в процессе поиска решения. На рисунке 1.5 показана начальная популяция потенциальных решений, которая равномерно покрывает область поиска решения. Далее явно видно, как постепенно с увеличением номера поколения особи "конденсируются" в окрестностях экстремумов и в конечном счете находится лучшее решение.



**Рис.1.6. Начальная «конденсация» особей популяции в окрестностях экстремумов**



**Рис.1.7. «Конденсация» особей в окрестностях экстремумов**

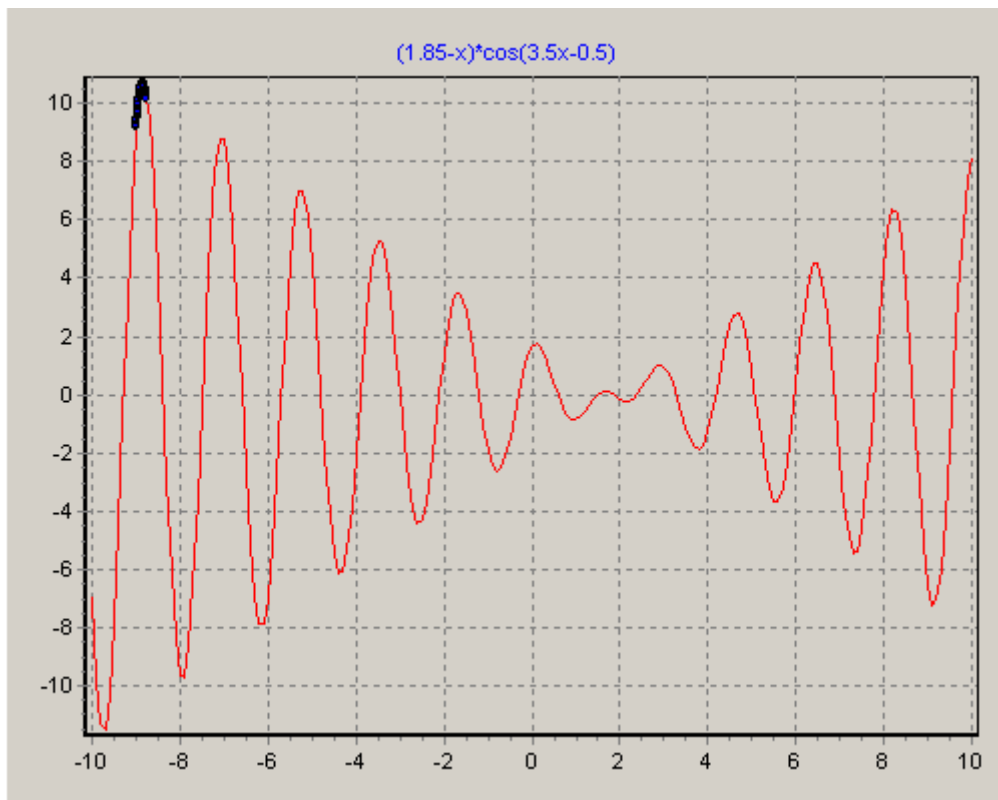


Рис.1.8. Положение особей популяции в конце эволюции

### Логарифмическое кодирование

Данный вид кодирования применяется для сокращения длины хромосом. При этом первый бит (a) кодовой последовательности используется для знака показательной функции, второй бит (b) – для знака степени этой функции, и остальные биты (str) представляют значение самой степени. Таким образом, двоичный код  $\langle a \ b \ str \rangle$  представляет вещественное число  $(-1)^b e^{(-1)^a [str]_{10}}$ . Здесь  $[str]_{10}$  означает десятичное число, представленное двоичным кодом str. Например, двоичный код  $\langle 10110 \rangle$  представляет вещественное число  $r = (-1)^0 e^{(-1)^1 [1110]_{10}} = e^{-6} = 0,002478752$ . Следует отметить, что при этом кодировании пять битов позволяет кодировать вещественные числа из интервала  $[-e^7, e^7]$ , что значительно больше, чем метод кодирования вещественных чисел, описанный выше.

## Порядок выполнения лабораторной работы

### 1. При домашней подготовке:

- изучить теоретический материал;
- ознакомиться с вариантами кодирования хромосомы;
- рассмотреть способы выполнения операторов репродукции, кроссинговера и мутации;
- выполнить индивидуальное задание на любом языке высокого уровня с необходимыми комментариями и выводами.

### 2. Во время занятия:

- продемонстрировать результаты выполнения работы;
- получить допуск к защите лабораторной работы.

### 3. Защитить отчет по лабораторной работе.

## Задание

1. Разработать простой генетический алгоритм для нахождения оптимума заданной по варианту функции одной переменной (таб. 1.1).

Вид экстремума:

Вариант	Вид экстремума
$\leq 15$	Максимум
$> 15$	Минимум

2. Исследовать зависимость времени поиска, числа поколений (генераций), точности нахождения решения от основных параметров генетического алгоритма:

- число особей в популяции
- вероятность кроссинговера, мутации.

3. Вывести на экран график данной функции с указанием найденного экстремума для каждого поколения.

4. Сравнить найденное решение с действительным.



Таблица 1.1. Индивидуальные задания.

Вариант	Вид функции	Промежуток нахождения решения
1	$(1.85-x) \cdot \cos(3.5x-0.5)$	$x \in [-10, 10]$
2	$\cos(\exp(x)) / \sin(\ln(x))$	$x \in [2, 4]$
3	$\sin(x) / x^2$	$x \in [3.1, 20]$
4	$\sin(2x) / x^2$	$x \in [-20, -3.1]$
5	$\cos(2x) / x^2$	$x \in [-20, -2.3]$
6	$(x-1) \cos(3x-15)$	$x \in [-10, 10]$
7	$\ln(x) \cos(3x-15)$	$x \in [1, 10]$
8	$\cos(3x-15) / \text{abs}(x)$ $=0$	$x \in [-10, -0.3), (0.3, 10]$ $x \in [-0.3, 0.3]$
9	$\cos(3x-15) \cdot x$	$x \in [-9.6, 9.1]$
10	$\sin(x) / (1 + \exp(-x))$	$x \in [0.5, 10]$
11	$\cos(x) / (1 + \exp(-x))$	$x \in [0.5, 10]$
12	$(\exp(x) - \exp(-x)) \cos(x) /$ $(\exp(x) + \exp(-x))$	$x \in [-5, 5]$
13	$(\exp(-x) - \exp(x)) \cos(x) /$ $(\exp(x) + \exp(-x))$	$x \in [-5, 5]$
14	$\cos(x-0.5) / \text{abs}(x)$	$x \in [-10, 0), (0, 10]$ , min
15	$\cos(2x) / \text{abs}(x-2)$	$x \in [-10, 2), (2, 10]$ , max

### Содержание отчета.

1. Титульный лист.
2. Индивидуальное задание по варианту.
3. Краткие теоретические сведения.

4. Программа и результаты выполнения индивидуального задания с комментариями и выводами.
5. Письменный ответ на контрольный вопрос по варианту (номер контрольного вопроса совпадает с номером варианта).

### **Контрольные вопросы**

1. Какие “источники” ГА?
2. Какие генетические операторы используются в ГА?
3. Какую роль в ГА играет оператор репродукции (ОР)?
4. Опишите реализацию ОР в виде колеса рулетки и приведите пример его работы.
5. Придумайте другую реализацию ОР.
6. Опишите 1-точечный оператор кроссинговера (ОК) и приведите пример его работы.
7. Придумайте другую реализацию ОК.
8. Какую роль играет оператор мутации (ОМ)?
9. Опишите ОМ и приведите пример его работы.
10. Придумайте другую реализацию ОМ.
11. Выполните программную реализацию простого ГА на одном из языков программирования для поиска максимума функции  $f(x) = 3x^3 + 2$  на интервале  $[-5, 5]$ .
12. Какие основные параметры ГА?
13. Исследуйте зависимость работы (скорость сходимости) ПГА от мощности популяции  $N$  (варьируя ее значение).
14. Исследуйте зависимость работы ПГА от значения вероятности ОК  $P_c$ .
15. Исследуйте зависимость работы ПГА от значения вероятности ОК  $P_m$ .