

# **ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ И ГЕНЕТИЧЕСКИЕ ОПЕРАТОРЫ**

---

**Массель Л.В., д.т.н., профессор  
кафедры Автоматизированных систем  
факультета Кибернетики ИрГТУ**

---

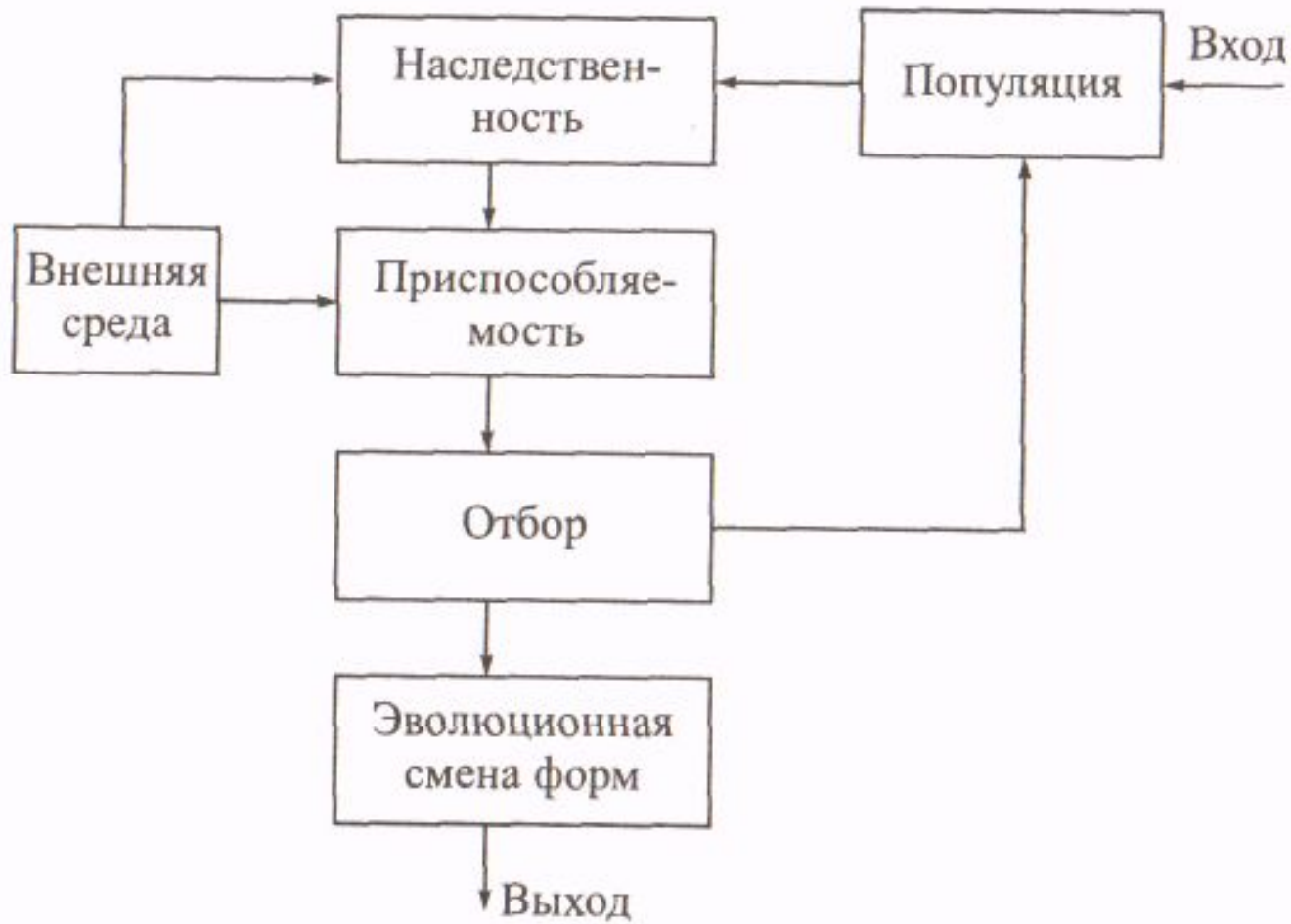
---

- *Основой* для возникновения генетических алгоритмов послужили модель биологической эволюции и методы случайного поиска. Л. Растригин отмечал, что случайный поиск возник как реализация простейшей модели эволюции, когда случайные мутации моделировались случайными шагами оптимального решения, а отбор — «устранением» неудачных вариантов.
- *Эволюционный поиск* с точки зрения преобразования информации — это последовательное преобразование одного конечного нечеткого множества промежуточных решений в другое.
- Само преобразование можно назвать алгоритмом поиска, или генетическим алгоритмом. *Генетические алгоритмы — это не просто случайный поиск.* Они эффективно используют информацию, накопленную в процессе эволюции.

# 1. МОДЕЛЬ ЭВОЛЮЦИИ ДАРВИНА



### 3. МОДЕЛЬ ЭВОЛЮЦИИ ЛАМАРКА



# ЦЕЛЬ ГА

---

- Цель генетических алгоритмов состоит в том, чтобы:
  - абстрактно и формально объяснять адаптацию процессов в естественной системе и интеллектуальной исследовательской системе;
  - моделировать естественные эволюционные процессы для эффективного решения оптимизационных задач науки и техники.
- В настоящее время используется новая парадигма решений оптимизационных задач на основе генетических алгоритмов и их различных модификаций. Генетические алгоритмы осуществляют поиск баланса между эффективностью и качеством решений за счет «выживания сильнейших альтернативных решений» в неопределенных и нечетких условиях.

- Генетические алгоритмы отличаются от других оптимизационных и поисковых процедур следующим:
  - работают в основном не с параметрами задачи, а с закодированным множеством параметров;
  - осуществляют поиск не путем улучшения одного решения, а путем использования сразу нескольких альтернатив на заданном множестве решений;
  - используют целевую функцию, а не ее различные приращения для оценки качества принятия решений;
  - применяют не детерминированные, а вероятностные правила анализа оптимизационных задач.

# ОТЛИЧИЯ ГА

---

- Для работы генетических алгоритмов выбирают множество натуральных параметров оптимизационной проблемы и кодируют их в последовательность конечной длины в некотором алфавите.
- Они работают до тех пор, пока не будет выполнено заданное число генераций (итераций алгоритма) или на некоторой генерации будет получено решение определенного качества, или когда найден локальный оптимум, т. е. возникла преждевременная сходимость и алгоритм не может найти выход из этого состояния.
- В отличие от других методов оптимизации эти алгоритмы, как правило, анализируют различные области пространства решений одновременно и поэтому они более приспособлены к нахождению новых областей с лучшими значениями целевой функции.

# ПРЕДВАРИТЕЛЬНЫЕ ЭТАПЫ ГА

---

- При решении практических задач с использованием генетических алгоритмов обычно выполняют четыре предварительных этапа:
  - **выбор способа представления** решения;
  - **разработка операторов** случайных изменений;
  - **определение способов** «выживания» решений;
  - **создание начальной популяции** альтернативных решений.



# ПЕРВЫЙ ПРЕДВАРИТЕЛЬНЫЙ ЭТАП ГА

---

- *На первом этапе* для представления решения в формальном виде требуется такая структура, которая позволит кодировать любое возможное решение и производить его оценку.
- Математически доказано, что не существует идеальной структуры представления, так что для создания хорошей структуры требуется анализ, перебор и эвристические подходы.
- Возможный вариант представления должен позволять проведение различных перестановок в альтернативных решениях.
- Для оценки решений необходимо определить способ вычисления целевой функции.

# ВТОРОЙ ПРЕДВАРИТЕЛЬНЫЙ ЭТАП ГА

---

- На втором этапе достаточно сложным является **выбор случайного оператора (или операторов) для генерации потомков**. Существует огромное число таких операторов. Существуют два основных типа размножения: половое и бесполое. При половом размножении два родителя обмениваются генетическим материалом, который используется при создании потомка. Бесполое размножение — это фактически клонирование, при котором происходят различные мутации при передаче информации от родителя к потомку. Модели этих типов размножения играют важную роль в генетических алгоритмах.
- В общем случае можно применить модели размножения, которые не существуют в природе. Например, использовать материал от трех или более родителей, проводить голосование при выборе родителей и т. п. При решении технических задач нет смысла слепо копировать законы природы и ограничиваться только ими.
- Успех генетических алгоритмов во многом зависит от того, **как взаимодействуют между собой схема представления, методы случайных изменений и способ определения целевой функции**. Поэтому для определенного класса задач целесообразно использовать направленные методы.
- В качестве примера рассмотрим два способа представления перестановок при решении оптимизационных задач. **В первом случае будем использовать одного родителя (альтернативное решение) и получать одного потомка**. Во втором случае используем **двух родителей, случайно выберем точку перестановки и для образования потомка возьмем первый сегмент у первого родителя, а второй сегмент — у второго**. Первый метод похож на бесполое размножение, а второй — на половое размножение. Стоит отметить, что если первый метод всегда генерирует реальное решение, то второй может генерировать недопустимые решения. При этом требуется «восстанавливать» допустимые решения перед их оценкой.

# ТРЕТИЙ И ЧЕТВЕРТЫЙ ПРЕДВАРИТЕЛЬНЫЕ ЭТАПЫ ГА

- **На третьем из рассматриваемых этапов задаются правила выживания решений для создания потомства.** Существует множество способов проведения селекции альтернативных решений. Простейшее правило — это «выживание сильнейших», т. е. остаются только лучшие решения с точки зрения заданной целевой функции, а все остальные устраняются. Такое правило часто оказывается малоэффективным при решении сложных технических проблем. Иногда лучшие решения могут происходить от худших, а не только от самых лучших. Тем не менее, логично использовать принцип: **«Чем «лучше» решение, тем больше вероятность его выживания»**
- **На последнем предварительном этапе создается начальная популяция.** При неполноте исходных данных о проблеме решения могут случайным образом выбираться из всего множества альтернатив. Это реализуется генерацией случайных внутрихромосомных перестановок, каждая из которых представляет собой определенное решение. **При создании начальной популяции рекомендуется использовать знания о решаемой задаче.** Например, эти знания могут быть получены из опыта разработчика, существующих стандартов и библиотек алгоритмов решения задач данного класса.

# ЭФФЕКТИВНОСТЬ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

---

**Эффективность генетического алгоритма — степень реализации запланированных действий алгоритма и достижение требуемых значений целевой функции.** Эффективность во многом определяется структурой и составом начальной популяции. При создании начального множества решений происходит **формирование популяции** на основе четырех основных принципов:

- «одеяло» — генерируется полная популяция, включающая все возможные решения в некоторой заданной области;
- «дробовик» — подразумевает случайный выбор альтернатив из всей области решений данной задачи.
- «фокусировка» — реализует случайный выбор допустимых альтернатив из заданной области решений данной задачи.
- «комбинирование» — состоит в различных совместных реализациях первых трех принципов.
- Отметим, что популяция обязательно является конечным множеством.

# ГЕНЕТИЧЕСКИЕ ОПЕРАТОРЫ

---

- В каждой генерации генетического алгоритма хромосомы являются **результатом применения некоторых генетических операторов**.
- **Оператор** — это языковая конструкция, представляющая один шаг из последовательности действий или набора описаний алгоритма.
- **Генетический алгоритм состоит из набора генетических операторов**.
- **Генетический оператор** по аналогии с оператором алгоритма - средство отображения одного множества на другое. Другими словами, это конструкция, представляющая один шаг из последовательности действий генетического алгоритма.

- **Оператор репродукции (селекции)**
- **Оператор кроссинговера**
- **Оператор мутации**
- **Оператор инверсии**

# ОПЕРАТОР РЕПРОДУКЦИИ (СЕЛЕКЦИИ)

---

- **Оператор репродукции (селекция)** — это процесс, посредством которого хромосомы (альтернативные решения), имеющие более высокое значение целевой функции (с «лучшими» признаками), получают большую возможность для воспроизводства (репродукции) потомков, чем «худшие» хромосомы. Элементы, выбранные для репродукции, обмениваются генетическим материалом, создавая аналогичных или различных потомков.
- Существует большое число видов операторов репродукции. Основным является:
  - **Селекция на основе рулетки** — это простой и широко используемый в простом генетическом алгоритме метод. При его реализации каждому элементу в популяции соответствует зона на колесе рулетки, пропорционально соразмерная с величиной целевой функции. Причем элемент с большим значением целевой функции имеет большую вероятность для выбора.

# ОПЕРАТОРЫ КРОССИНГОВЕРА (СКРЕЩИВАНИЯ).

---

**Оператор кроссинговера** — это языковая конструкция, позволяющая на основе преобразования (скрещивания) хромосом родителей (или их частей) создавать хромосомы потомков. Существует огромное число операторов кроссинговера, так как их структура в основном и определяет эффективность генетических алгоритмов.

**Простой (одноточечный) оператор кроссинговера.** Перед началом работы одноточечного оператора кроссинговера определяется так называемая **точка оператора кроссинговера, или разрезающая точка** оператора кроссинговера, которая обычно определяется случайно. Эта точка определяет место в двух хромосомах, где они должны быть «разрезаны».

Например, пусть популяция  $P$  состоит из хромосом  $P1$  и  $P2$ , которые выступают в качестве родителей,  $P = \{P1, P2\}$ . Пусть первый и второй родители имеют вид  $P1 : 11111$ ,  $P2 : 00000$ .

Выберем точку оператора кроссинговера между вторым и третьим генами в  $P1, P2$ .

Тогда, меняя элементы после точки оператора кроссинговера между двумя родителями, можно создать два новых потомка. В нашем примере получим:



# ОДНОТОЧЕЧНЫЙ ОПЕРАТОР КРОССИНГОВЕРА

---

$$P_1 : \quad 1 \quad 1 \quad | \quad 1 \quad 1 \quad 1$$

$$P_2 : \quad 0 \quad 0 \quad | \quad 0 \quad 0 \quad 0$$

---

$$P'_1 : \quad 1 \quad 1 \quad | \quad 0 \quad 0 \quad 0$$

$$P'_2 : \quad 0 \quad 0 \quad | \quad 1 \quad 1 \quad 1$$

# ОДНОТОЧЕЧНЫЙ ОПЕРАТОР КРОССИНГОВЕРА

---

- После применения оператора кроссинговера имеем две старые хромосомы и всегда получаем две новые хромосомы.
- Схематически простой оператор кроссинговера показывает преобразование двух хромосом и частичный обмен информацией между ними, использующий точку разрыва, выбранную случайно.

# ДВУХТОЧЕЧНЫЙ ОПЕРАТОР КРОССИНГОВЕРА

---

$$P_1 : \quad 1 \quad 1 \quad 1 \quad | \quad 0 \quad 1 \quad | \quad 0 \quad 0$$

$$P_2 : \quad 0 \quad 0 \quad 0 \quad | \quad 1 \quad 1 \quad | \quad 1 \quad 0$$

---

$$P'_1 : \quad 1 \quad 1 \quad 1 \quad | \quad 1 \quad 1 \quad | \quad 0 \quad 0$$

$$P'_2 : \quad 0 \quad 0 \quad 0 \quad | \quad 0 \quad 1 \quad | \quad 1 \quad 0$$

# ОПЕРАТОР МУТАЦИИ

---

**Оператор мутации** — языковая конструкция, позволяющая на основе преобразования родительской хромосомы (или ее части) создавать хромосому потомка.

Оператор мутации обычно состоит из двух этапов:

- 1. В хромосоме  $A = (a_1, a_2, a_3, \dots, a_{L-2}, a_{L-1}, a_L)$  определяются случайным образом две позиции (например,  $a_2$  и  $a_{L-1}$ ).
- 2. Гены, соответствующие выбранным позициям, переставляются, и формируется новая хромосома  $A' = (a_1, a_{L-1}, a_3, \dots, a_{L-2}, a_2, a_L)$
- Простейшим оператором мутации является **одноточечный**. При его реализации случайно выбирают ген в родительской хромосоме и, обменивая его на рядом расположенный ген, получают хромосому потомка

# ОДНОТОЧЕЧНЫЙ ОПЕРАТОР МУТАЦИИ

---

$$\begin{array}{l} P_1 : \quad 0 \quad 1 \quad 1 \quad | \quad 0 \quad 1 \quad 1 \\ P'_1 : \quad 0 \quad 1 \quad 0 \quad | \quad 1 \quad 1 \quad 1 \end{array}$$

# ДВУХТОЧЕЧНЫЙ ОПЕРАТОР МУТАЦИИ

---

$P$ :  $A \mid B \ C \ D \mid E \ F$   
 $P'$ :  $A \quad E \ C \ D \quad B \ F$

# ОПЕРАТОР ИНВЕРСИИ

---

**Оператор инверсии** - это языковая конструкция, позволяющая на основе инвертирования родительской хромосомы (или ее части) создавать хромосому потомка. При его реализации случайным образом определяется одна или несколько точек разреза (инверсии), внутри которых элементы инвертируются.

- Генетический оператор инверсии состоит из следующих шагов.
  1. Хромосома  $V = b_1, b_2, \dots, b_L$  выбирается случайным образом из текущей популяции.
  2. Два числа  $y_1'$  и  $y_2'$  выбираются случайным образом из множества  $\{0, 1, 2, \dots, L + 1\}$ , причем считается, что  $y_1 = \min \{y_1', y_2'\}$  и  $y_2 = \max \{y_1', y_2'\}$ .
  3. Новая хромосома формируется из  $V$  путем инверсии сегмента, который лежит справа от позиции  $y_1$  и слева от позиции  $y_2$  в хромосоме  $V$ . Тогда после применения оператора инверсии получаем:

# ОДНОТОЧЕЧНЫЙ ОПЕРАТОР ИНВЕРСИИ

---

$P_2$  : A | B C D E F G H

$P'_2$  : A | H G F E D C B



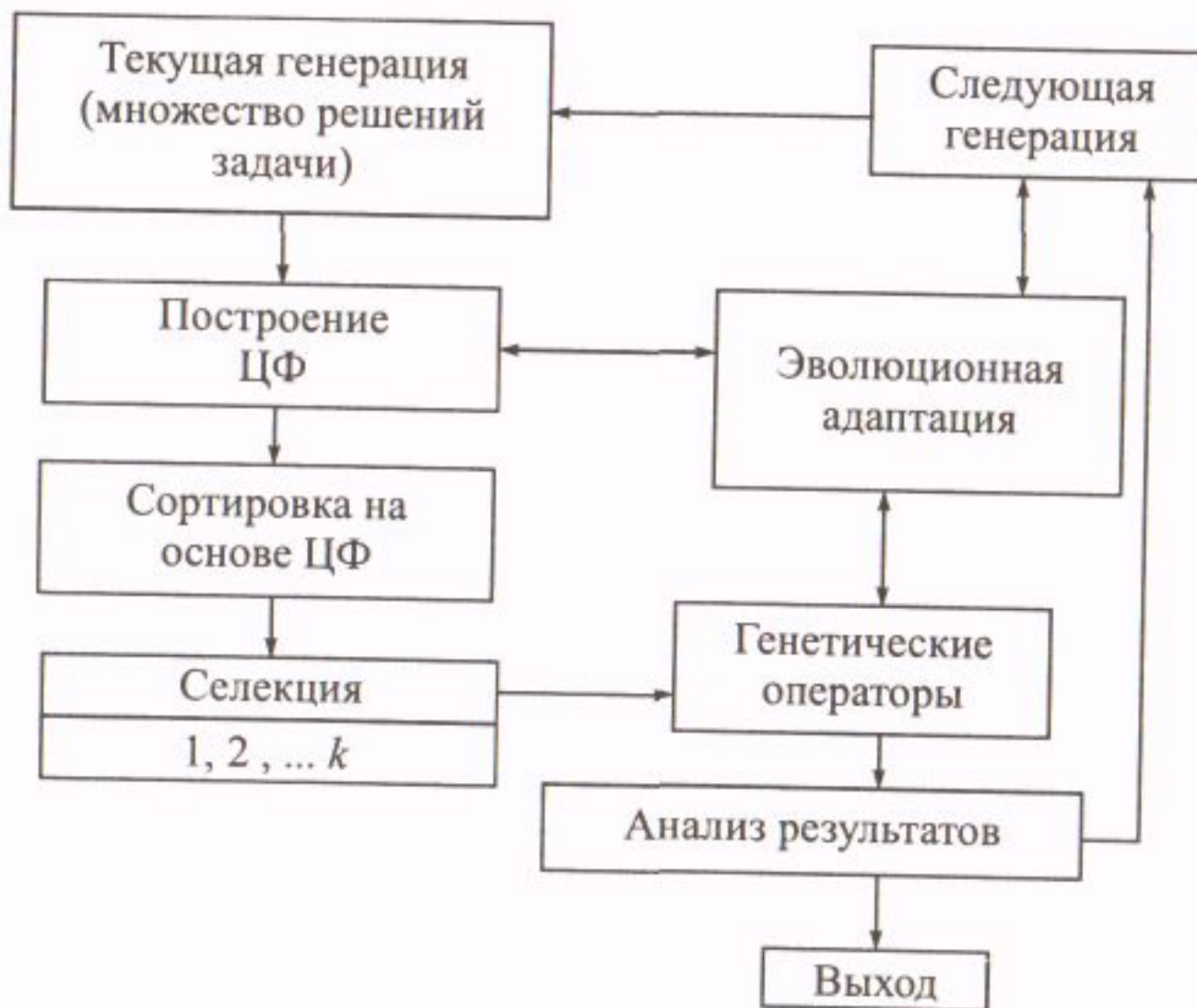
# ДВУХТОЧЕЧНЫЙ ОПЕРАТОР ИНВЕРСИИ

---

$$P_1 : A \ B \ C \ | \ D \ E \ F \ | \ G \ H$$
$$P'_1 : A \ B \ C \ | \ F \ E \ D \ | \ G \ H$$

- ❑ Транслокации
- ❑ Транспозиции
- ❑ Сегрегации
- ❑ Удаления
- ❑ Вставки
- ❑ Редукции
- ❑ Рекомбинации

- *Простой генетический алгоритм* случайно генерирует популяцию хромосом (альтернативных упорядоченных и неупорядоченных решений). Затем производится копирование хромосом, перестановка их частей и генерация новых хромосом (решений) на основе трех операторов: репродукции, кроссинговера и мутации.



# ОСНОВНЫЕ ЭТАПЫ ЭВОЛЮЦИОННОГО ПОИСКА (простой генетический алгоритм Холланда)

---

1. **Сконструировать начальную популяцию.** Ввести точку отсчета по коленим  $t = 0$ . Вычислить приспособленность каждой хромосомы в популяции, а затем среднюю приспособленность всей популяции.
2. Установить  $t = t+1$ . **Произвести выбор двух родителей (хромосом)** для реализации оператора кроссинговера. Он выполняется случайным образом пропорционально приспособляемости родителей.
3. Сформировать генотип потомков. Для этого с заданной вероятностью выполнить **оператор кроссинговера над генотипами выбранных хромосом**. Далее с вероятностью 0,5 выбрать один из потомков  $P_i(t)$  и сохранить как член новой популяции. После этого к  $P_i(t)$  последовательно применить оператор инверсии, а затем — оператор мутации с заданными вероятностями. Полученный генотип потомка сохранить как  $P_k(t)$ .
4. **Определить количество хромосом для исключения их из популяции**, чтобы ее размер оставался постоянным. Текущую популяцию обновить заменой отобранных хромосом на потомков  $fit(t)$ .
5. **Произвести определение приспособленности (целевой функции)** и пересчет средней приспособленности всей полученной популяции.
6. Если  $t =$  Заданному, то **перейти к 7**, если нет, то **перейти к 2**.
7. Конец работы.

Данный алгоритм известен как **упрощенный «репродуктивный план Д.Холланда»**. Заметим, что в практических задачах вместо понятия «приспособленность» используют понятие «целевая функция».

# ПРОСТОЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ Гольдберга

---

*Простой генетический алгоритм* был впервые описан Д. Гольдбергом на основе работ Д.Холланда. Его механизм несложен. Предварительно простой генетический алгоритм случайно **генерирует популяцию последовательностей — хромосом** (альтернативных упорядоченных и неупорядоченных решений). Затем производится **копирование последовательности хромосом и перестановка их частей**. Далее простой генетический алгоритм реализует множество простых операций к начальной популяции и генерирует новые решения.

Простой генетический алгоритм состоит из трех операторов:

- **репродукции;**
- **кроссинговера;**
- **мутации.**

**Репродукция** — процесс, в котором хромосомы копируются пропорционально значению их целевой функции. Копирование хромосом с «лучшим» значением целевой функции имеет большую вероятность для попадания в следующую генерацию. Рассматривая эволюцию Дарвина, можно отметить, что **оператор репродукции** является искусственной версией натуральной селекции — «выживание сильнейших». Он представляется в алгоритмической форме различными способами. Самый простой — **создать модель «колеса рулетки»**, в которой каждая хромосома имеет поле, пропорциональное значению целевой функции.

- Рассмотрим пример : необходимо найти значение максимума функции  $f(x) = x^2$  на целочисленном интервале  $[0,31]$ . Традиционными методами можно изменять значения переменной  $x$ , пока не получим максимальное значение  $f(x)$ .
- Для объяснения и реализации простого генетического алгоритма построим следующую таблицу

# ТАБЛИЦА 1

---

Номер хромосом	Хромосома		Значение ЦФ	Значение ЦФ, в процентах
	(двоичный код)	Десятичный код		
1	0 1 1 0 0	12	144	16,2
2	1 0 0 0 0	16	256	28,8
3	0 0 1 1 1	7	49	5,5
4	1 0 1 0 1	21	441	49,5

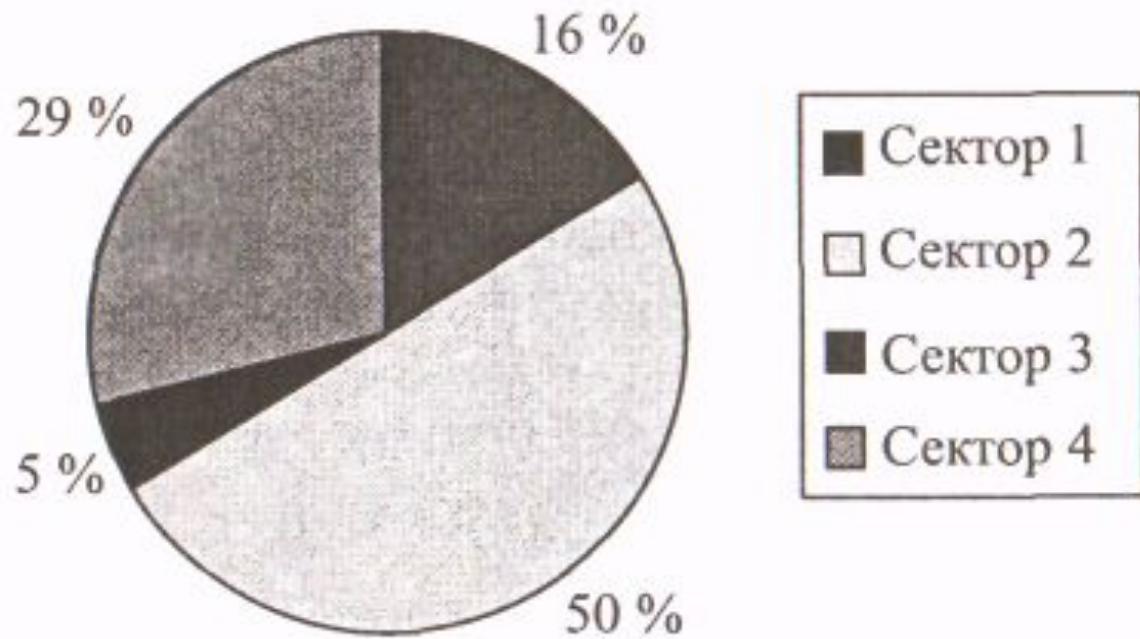


# КОММЕНТАРИЙ К ТАБЛИЦЕ 1

---

- В столбце 2 таблицы расположены 4 хромосомы (представленные в двоичном коде), сгенерированные случайным образом. Значение целевой функции для каждой хромосомы (столбец 4) определяется как квадрат значения десятичного кода двоичного числа, которое приведено для хромосом в третьем столбце таблицы. Тогда суммарное значение целевой функции всех хромосом равно 890. Для селекции хромосом используется оператор репродукции на основе колеса рулетки.
- Для упрощения в рассматриваемом примере будем считать, что  $16,2 = 16$ ;  $49,5 = 50$ ;  $5,5 = 5$ ;  $28,8 = 29$

# КОЛЕСО РУЛЕТКИ



- Для селекции хромосом используется **оператор репродукции на основе колеса рулетки**.
- На предыдущем слайде поля колеса рулетки соответствуют значению целевой функции каждой хромосомы в процентах. В одной генерации колесо рулетки вращается, и после остановки ее указатель определяет хромосому, выбранную для реализации следующего оператора. Очевидно, не всегда хромосома с большим значением целевой функции в результате применения оператора репродукции будет выбрана для дальнейших преобразований.

# ОПЕРАТОР КРОССИНГОВЕРА

---

- На основе реализации **оператора репродукции** выбираются хромосомы для применения **оператора кроссинговера**.
- Оператор кроссинговера, как правило, выполняется в 3 шага, одним из операторов кроссинговера, описанным выше. Точка разрыва  $k$  выбирается случайно между 1 и числом, равным длине хромосомы минус единица, т. е. в интервале  $(1, L - 1)$ . Длина хромосомы  $L$  — это число значащих цифр в ее коде.
- В рассматриваемом примере длина каждой хромосомы равна пяти ( $L = 5$ ). На основе оператора кроссинговера создаются две новые хромосомы путем обмена их частей между позициями  $(k + 1)$  и  $L$  соответственно.
- Например, рассмотрим хромосомы 1 и 2 из начальной популяции (табл. 3.1). Пусть  $k = 1$ . Тогда получим:  
P1: 0 | 1 1 0 0 — до применения оператора  
P2: 1 | 0 0 0 0 кроссинговера,  

---

P1': 0 0 0 0 0 — после применения оператора  
P2': 1 1 1 0 0 кроссинговера

- 
- Работа простого генетического алгоритма начинается с репродукции. Хромосомы для следующей генерации выбираются путем вращения колеса рулетки. В примере колесо рулетки вращается 4 раза. Это число соответствует мощности начальной популяции.
  - Величину отношения  $f_i(x)/\sum f(x)$  называют вероятностью выбора копий (хромосом) при реализации оператора  $P_i(OP)$

# ВЕРОЯТНОСТЬ ВЫБОРА КОПИЙ (ХРОМОСОМ)

---

$$P_i(\text{OP}) = \frac{f_i(x)}{\text{sum } f(x)},$$

# ВЕРОЯТНОСТЬ ВЫБОРА КОПИЙ (ХРОМОСОМ)

---

где

$f_i(x)$  — значение целевой функции  $i$ -й хромосомы в популяции,

$\sum f(x)$  — суммарное значение целевой функции всех хромосом в популяции,

OP — оператор репродукции.

Величину  $P_i(OP)$  также называют нормализованной вероятностью выбора.

Ожидаемое число копий  $i$ -й хромосомы после реализации оператора репродукции определяется по формуле:

# ЧИСЛО КОПИЙ $i$ -Й ХРОМОСОМЫ

---

$$N_i = P_i(OP) \cdot N_G,$$



# ЧИСЛО КОПИЙ $i$ -Й ХРОМОСОМЫ

---

- где  $N_G$  — число анализируемых хромосом
- Ожидаемое число копий хромосомы  $P_i$ , переходящее в следующее поколение, также иногда определяют на основе выражения:

$$\langle \quad \rangle,$$

где  $f(x)$  с чертой — среднее значение целевой функции по всей популяции.

# ОЖИДАЕМОЕ ЧИСЛО КОПИЙ ХРОМОСОМЫ $P_i$

---

$$N_i = \frac{f_i(x)}{\bar{f}(x)},$$

## ПРИМЕР (ТАБЛИЦА 2)

---

- Тогда для рассматриваемого примера ожидаемое число копий для первой хромосомы из таблицы 1 равно  $0,16 \cdot 4 = 0,64$  копий, для второй —  $0,29 \cdot 4 = 1,16$  копий, для третьей —  $0,05 \cdot 4 = 0,2$  и, наконец, для четвертой —  $0,5 \cdot 4 = 2$ .
- Используя модель «бросание монеты», можно определить число полученных копий. Например, (см. столбец 7 табл. 2) хромосомы *P1* и *P2* получают 1 копию, хромосома *P4* - 2 копии и хромосома *P3* не получает копий. Сравнивая этот результат с ожидаемым числом копий, получаем, что «лучшие» хромосомы дают большее число копий, «средние» остаются и «плохие» удаляются после реализации оператора репродукции.

## ТАБЛИЦА 2

№	Начальная популяция	$x$	$f(x)$	Значение $f_i(x)/\text{sum}[f(x)]$	Ожидаемое число копий	Число полученных копий
1	0 1 1 0 0	12	144	0,16	0,65	1
2	1 0 0 0 0	16	256	0,29	1,15	1
3	0 0 1 1 1	7	49	0,05	0,22	0
4	1 0 1 0 1	21	441	0,55	1,98	2
Суммарное значение ЦФ ( $\text{sum } f(x)$ )			890	1,00	4,00	4
Среднее значение ЦФ $\bar{f}(x)$			222,5	0,22	0,88	1
Мах значение ЦФ			441	0,5	2	2

# ТАБЛИЦА 3

№	Популяция после оператора репродукции	Пары, выбранные случайно	Точка оператора кроссинговера	Новая популяция	$x$	$f(x)$	
1	0   1 1 0 0	1-4	1	0 0 1 0 1	5	25	
2	1 0   0 0 0	2-3	2	1 0 1 1 1	23	529	
3	0 0   1 1 1	2-3	2	0 0 0 0 0	0	0	
4	1   0 1 0 1	1-4	1	1 1 1 0 0	28	784	
Sum $f(x)$						1338	
$\bar{f}(x)$						334,5	
Max $f(x)$						784	

## КОММЕНТАРИИ К ТАБЛИЦЕ 3

---

- В столбце 2 приведен вид четырех хромосом после выполнения оператора репродукции.
- В столбце 3 приведены списки пар хромосом, которые выбраны случайным образом для реализации оператора кроссинговера.
- В столбце 4 указан номер позиции для точки разреза хромосом.
- В столбце 5 приведен вид 4 хромосом после выполнения оператора кроссинговера.
- В столбце 6 приведены значения десятичного кода двоичного числа каждой хромосомы столбца 5.
- В столбце 7 приведено значение целевой функции  $f(x)$  для каждой из 4 хромосом новой популяции.
- В строке 5 приведено суммарное значение ЦФ (целевой функции) хромосом новой популяции,
- в строке 6 — среднее значение их целевой функции,
- в строке 7 - максимальное значение целевой функции хромосомы из новой популяции.

- 
- Применяя к популяции, полученной после реализации оператора репродукции (столбец 2 табл. 3), оператор кроссинговера, получим новую популяцию хромосом (5-й столбец таблицы 3). В принципе оператор кроссинговера можно применять любое число раз. После проведения одной генерации простого генетического алгоритма улучшились все показатели: среднее и максимальное значение целевой функции.

# ОПЕРАТОР МУТАЦИИ

---

- Далее, согласно схеме выполнения простого генетического алгоритма, реализуется оператор мутации. Существует большое количество видов операторов мутации. Эти операторы соответствуют перестановкам элементов внутри заданного множества. Очевидно, что при небольшой длине хромосомы  $L$  (порядка 10-20) можно выполнить полный перебор за приемлемое время и найти наилучшие решения.
- При увеличении  $L$  до 50-200 и выше полный перебор произвести затруднительно и необходимы другие механизмы поиска. Здесь как раз и приходит на помощь направленно-случайный поиск, который реализуется на основе простого генетического алгоритма.



# ОПЕРАТОР МУТАЦИИ

---

- Применим, например, оператор мутации к хромосоме  $P2$  с ЦФ = 23 (столбец 5 табл. 3). Выберем позиции 2, 3 и после оператора мутации получим:

$P2 : 10| 1 1 1$

$P2' : 11 0 1 1$

- Снова, применяя оператором мутации к  $P2'$  между позициями 3 и 4, получим:

$P2' : 1 1 0| 1 1$

$P2'' : 1 1 1 0 1$

- Хромосома  $P2''$  имеет значение ЦФ = 29.
  - Опять применим к хромосоме  $P2''$  оператор мутации. Тогда, при случайном выборе точки оператора мутации между 4 и 5, получим:
- $P2'' : 1 1 1 0 ' 1$
- $P2''' : 1 1 1 1 ' 0$
- Хромосома  $P2'''$  имеет значение ЦФ = 30.
  - Итак, найдено квазиоптимальное значение  $x$ , равное 30 для решения задачи поиска максимального значения функции  $f(x) = x^1$  на интервале  $[0,31]$ . Заметим, что для хромосомы  $P2$  лучшего значения целевой функции, чем для  $P2'''$ , найти невозможно.

# ДРУГОЙ СТАНДАРТНЫЙ ТИП ГЕНЕТИЧЕСКОГО АЛГОРИТМА, ОПИСАННЫЙ Л.ДЕВИСОМ

---

1. Инициализировать популяции хромосом.
  2. Оценить значения каждой хромосомы в популяции.
  3. Создать новые хромосомы посредством скрещивания текущих хромосом; применить операторы мутации и рекомбинации.
  4. Устранить хромосомы из популяции, чтобы освободить место для новых хромосом.
  5. Оценить значения новых хромосом и вставить их в популяцию.
  6. Если время, заданное на реализацию алгоритма, закончено, то остановиться и возвратиться к наилучшей хромосоме; если нет, то перейти к 3.
  7. Конец работы алгоритма.
- Сравнивая описания простых генетических алгоритмов Д. Голдберга, Д. Холланда и Л.Девиса, видим, что в них реализована одна основная идея моделирования эволюции с некоторыми модификациями. Однако заметим, что эти изменения могут существенно влиять на окончательное качество решения.

# МОДИФИЦИРОВАННЫЙ ПРОСТОЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

---

1. Создать начальную популяцию решений.
2. Смоделировать популяцию (определить ЦФ (целевую функцию) для каждой хромосомы).
3. Если еще не проведено необходимое число генераций или не закончилось время, заданное на реализацию алгоритма, или не найдено оптимальное значение целевой функции (если оно известно):
  - а) выбрать элементы для репродукции;Применить:
  - б) оператор кроссинговера для создания потомков;
  - в) оператор мутации;
  - г) оператор инверсии;
  - д) оператор транспозиции;
  - е) оператор транслокации;
  - ж) оператор сегрегации;
  - з) оператор удаления вершин;
  - и) оператор вставки вершин;
  - к) рекомбинацию родителей и потомков для создания новой генерации;
  - л) оператор редукции.
4. Реализовать новую генерацию.

---

**Благодарю за внимание!**