

## Лекция Генетические алгоритмы

Эволюционное моделирование (evolutionary computation) - направление в искусственном интеллекте, в основе которого лежат принципы и понятийный

аппарат, заимствованные из эволюционной биологии и популяционной генетики и объединяющие компьютерные методы моделирования эволюционных процессов в искусственных системах

### **Методы эволюционного моделирования**

генетические алгоритмы

генетическое программирование

эволюционное программирование и эволюционные стратегии

### **Области применения эволюционного моделирования**

автоматизации решения оптимизационных задач науки и техники

изучения и моделирования процессов естественной эволюции свойствами адаптивного поведения и самоорганизации на основе методов эволюционного моделирования

Оптимизационные задачи затрагивают максимизацию или минимизацию некоторой целевой функции на множестве ограничений

**Простейшая оптимизационная задача – линейное программирование**, пространство решений ограничено множеством линейных ограничений и целевая функция имеет линейный вид. Переменные должны быть положительными величинами. В этом случае оптимальное решение обычно лежит в точках пересечения границ

**Линейное программирование** снижает размерность комбинаторной задачи:

- вместо всех точек области допустимых решений проверяются только ее вершины
- число ограничений очень большое
- целевая функция или ограничения нелинейные
- пространство решений включает только дискретные значения

### **Оптимизационные задачи становятся более сложными**

Если целевая функция нелинейная, то оптимальное решение может существовать как внутренняя точка пространств поиска, т.е. лежать не только в вершинах области допустимых решений.

### **Особенности генетических алгоритмов**

- Используют не непосредственно параметры задачи, а их закодированный вид
- Ведут поиск, который исходит не из единственной точки, а из полной популяции (т.е. множества точек)
- Используют только функцию оценки и никакой другой вспомогательной информации
- Применяют вероятностные, а не детерминированные правила выбора
- В процессе поиска используется значение целевой функции, а не ее приращения
- Выполняется одновременный анализ различных областей пространства решений, в связи с чем возможно нахождение новых областей с лучшими значениями целевой функции за счет объединения субоптимальных решений
- из разных популяций

### **Преимущества**

- В алгоритм легко внедрить новые операторы
- Проще для организации параллельных вычислений
- Кодирование исходных параметров, работа на популяциях, с группой решений, стохастические действия дают эффект устойчивости (высокая помехозащищенность)

- Независимость от вида функции, включая поддержку неаналитического задания функции
- Независимость от области определе

Эволюционная модель	Математическая модель
Хромосома	Решение, объект, строка, последовательность
Ген	Переменная, параметр, характеристика, признак
Аллель	Значение фрагмента закодированного параметра
Локус	Номер фрагмента закодированного параметра
Генотип	Множество закодированных решений задачи, пространство поиска
Фенотип	Множество решений задачи, пространство решений
Особь, индивидуум	Объект, система
Пригодность, приспособленность	Качество, оптимальность
Fitness-функция	Целевая функция
Популяция	Множество решений
Поклоение	Итерация работы эволюционного алгоритма

Хромосома	Строка
Ген	Значение функции, характеристика, детектор
Аллель	Возможные значения генов
Локус	Позиция в строке
Генотип	Фактическая структура. Кодированная хромосома.
Фенотип	Множество параметров, альтернативное решение, декодированная структура
Эпистаз	Нелинейность. Взаимодействие генов.

**Генетический алгоритм** – последовательность управляющих действий и операций, моделирующая эволюционные процессы на основе аналогов механизмов генетического наследования и естественного отбора. Степень адаптации зависит от набора хромосом, полученных от родителей

**Дж. Холланд (1975)** работал над алгоритмами, оперирующими рядами бинарных цифр

**Популяция** - множество индивидуумов определенной численности

**Индивидуум** в популяции - закодированное в виде хромосом (кодовых рядов, генов) множество параметров задачи

**Ген** - признак, бит, единичный элемент индивидуума

**Фенотип** - множество параметров задачи (решения, точки пространства поиска), представленные не в кодированном битами (т. е. на уровне генотипа), как правило, десятичном виде

**Функция приспособляемости, функция оценки** отражает приспособляемость индивидуума в популяции. По результатам оценки нужно выбрать индивидуума лучше всего приспособленного, т.е. с наибольшим значением функции оценки. Согласно принципу эволюции, выживает сильнейший и лучше приспособленный

**Хромосома** – вектор (последовательность) из нулей и единиц, каждая позиция (бит) которого называется геном. Отдельные гены в хромосомах рассматриваются как уникальные переменные

**Особь (индивидуум)** – набор хромосом, содержащий генетический код

Используется явное разделение на пространство поиска (генотип) и пространство решений (фенотип)

Каждое решение кодируется в виде бинарной хромосомы  $S$  длиной  $L(S)$ , состоящей из  $n$  числа генов  $g$

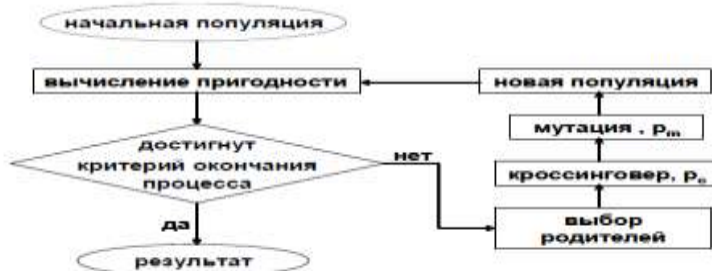
Каждый ген есть двоичный код длиной  $L(g)$ , соответствующий одной переменной задачи оптимизации

- использование кода Грея

- символьное
- вещественных чисел
- определяется программистом

**Теорема Холланда:** при отборе «здоровых» хромосом в популяции остаются только те, которые лучше всех приспособлены к окружающей среде

### Простой генетический алгоритм



**Инициализация** – создание начальной популяции (производится произвольно). Популяция должна быть разнообразной. Допускается добавлять в популяцию «здоровые» хромосомы

$$r = \frac{g(\max - \min)}{(2^n - 1)} + \min, \quad g = \frac{(r - \min)}{(\max - \min)(2^n - 1)}$$



- В случае двоичного кодирования используется n бит для каждого параметра
- g-значение параметра в двоичном формате
- r-значение параметра с плавающей точкой

Декодирование фрагментов хромосом в процент востри переменных

Код Грея	Двоично-десятичный вид	Десятичное значение числа	Вещественное значение координаты
0000	0000	0	a
0001	0001	1	a+1(b-a)/15
0011	0010	2	a+2(b-a)/15
0010	0011	3	a+3(b-a)/15
0110	0100	4	a+4(b-a)/15
0111	0101	5	a+5(b-a)/15
0101	0110	6	a+6(b-a)/15
0100	0111	7	a+7(b-a)/15
1100	1000	8	a+8(b-a)/15
1101	1001	9	a+9(b-a)/15
1111	1010	10	a+10(b-a)/15
1110	1011	11	a+11(b-a)/15
1010	1100	12	a+12(b-a)/15
1011	1101	13	a+13(b-a)/15
1001	1110	14	a+14(b-a)/15
1000	1111	15	b

### Генетические операторы

- Оператор отбора (селекции)
- Оператор кроссинговера (рекомбинации)
- Оператор мутации
- Оператор инверсии

#### Оператор отбора (селекции)

Наиболее важный этап в работе генетических алгоритмов. Хромосома отбирается для дальнейшего использования в другой популяции

Поэтому выбирается группа хромосом - если выбрать только «здоровые» хромосомы, то решение проблемы становится ограниченным.

#### Методы отбора хромосом (селекция)

**Метод вероятностного выбора:** чем выше здоровье хромосомы, тем больше вероятность ее выбора для формирования следующего поколения популяции. Наиболее популярный

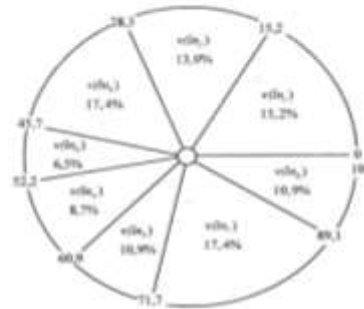
**Метод элиты:** автоматический перенос некоторого количества самых здоровых хромосом в следующее поколение (например, 10%)

**Метод турнира:** отбираются две или три хромосомы, которые затем соревнуются за право попасть в следующее поколение

$$v(In_i) = p(In_i) \cdot 100\%$$

$$p(In_i) = \frac{F(In_i)}{\sum_{i=1}^N F(In_i)}$$

$In$  – индивидум,  $n$  – численность популяции  
 $v(In)$  – сектор колеса рулетки  
 $F(In)$  – значение функции оценки индивидуума  
 $p(In)$  – вероятность выбора индивидуума



Применяется для определения на основе значений **fitness-функции хромосом-кандидатов** в следующее поколение. В генетическом алгоритме могут быть использованы различные схемы селекции

**Пропорциональный отбор**, когда число копий хромосомы пропорционально ее

- оптимальности. В следующее поколение могут перейти хромосомы только с
- оптимальностью выше средней.

**Отбор на основе «колеса рулетки».** Чем выше оптимальность хромосомы, тем

- больше её сектор на колесе рулетки. Случайная составляющая этого метода отбора
- дает шанс всем хромосомам попасть в следующее поколение

**Турнирный отбор.** Популяция случайно разбивается на группы из  $N_t$  хромосом. Из

- каждой группы лучшая хромосома выбирается в следующее поколение. Самая
- «худшая» хромосома популяции не имеет шансов попасть в следующее поколение

**Отбор на основе ранжирования** (линейного, равномерного) членов популяции по их

- приспособленности. На основе ранга хромосомы вычисляется вероятность ее
- попадания в следующую популяцию

**Для гарантированного попадания** лучшей хромосомы в следующее поколение

- используется стратегия элитизма

**Этап рекомбинирования - кроссинговер - кроссовер**

**Кроссовер** – перекрестное скрещивание, части хромосом изменяются и перемещаются, полученные новые хромосомы включаются для формирования следующего поколения

**Рекомбинирование производится до тех пор** пока не будет решена задача или выполнено определенное условие (например, максимально возможное количество поколений)

Перекрестное скрещивание и мутация являются прямыми аналогами естественных процессов

**Кроссовер, кроссинговер, перекрестное скрещивание (crossover)** – операция, при которой две хромосомы обмениваются своими частями, производится при участии двух хромосом

Оператор делит их в произвольной точке для каждой хромосомы и меняет местами части хромосом. Разделение может производиться в нескольких точках. Образуются две новые хромосомы

При перекрестном скрещивании в популяции изменяется существующий набор хромосом с целью создания новых. Это позволяет генетическому алгоритму выполнять **поиск решения проблемы среди существующих решений**

**Оператор мутации состоит в случайном изменении (на противоположное)**

- значения каждого бита с некоторой (обычно малой) вероятностью  $P_m$ .

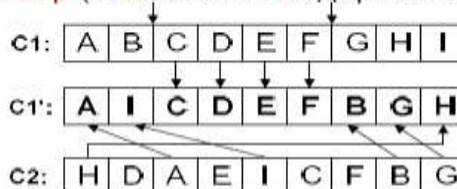
**Оператор инверсии участок хромосомы в своем поле разворачивается на 180 град.**

- разворачивается на 180 градусов

**Одноточечный кроссинговер (бинарное кодирование)**



**Упорядоченный кроссинговер (символьное кодирование)**



**Арифметический кроссинговер (вещественное кодирование)**

**Другие модификации кроссинговера для различных видов кодирования**

**Условия окончания работы генетического алгоритма**

- Достижение заданного числа поколений
- Снижение разнообразия популяции и ее вырождение в подавляющее большинство одинаковых по приспособленности хромосом
- Снижение скорости сходимости алгоритма (на протяжении определенного числа поколений качество решений не изменяется)
- Получение решения, удовлетворяющего пользователя
- Достижение лимита затраченного на поиск времени, числа выполнения определенных фрагментов алгоритма
- В качестве результата работы ГА принимается хромосома последнего поколения, имеющая самое лучшее значение fitness-функции

Основные термины и определения

Глоссарий

Термины и определения. Генетические алгоритмы

(См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Вектор — упорядоченный набор чисел, называемых компонентами вектора. Так как вектор можно представить в виде строки его координат то в дальнейшем понятия вектора и строки считаются идентичными. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Булев вектор — вектор, компоненты которого принимают значения из двух элементного (булева) множества, например,  $\{0, 1\}$  или  $\{-1, 1\}$ . Хеммингово расстояние — используется для булевых векторов и равно числу различающихся в обоих векторах компонент. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Хеммингово пространство — пространство булевых векторов, с введенным на нем расстоянием (метрикой) Хемминга. В случае булевых векторов размерности  $n$  рассматриваемое пространство представляет собой множество вершин  $n$ -мерного гиперкуба с хемминговой метрикой. Расстояние между двумя вершинами, определяется длиной кратчайшего соединяющего их пути, измеренной вдоль ребер. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Хромосома — вектор (или строка) из каких-либо чисел. Если этот вектор представлен бинарной строкой из нулей и единиц, например, 1010011, то он получен либо с использованием двоичного кодирования, либо кода Грея. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Каждая позиция (бит) хромосомы называется геном. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Индивидуум (генетический код, особь) — набор хромосом (вариант решения задачи). Обычно особь состоит из одной хромосомы, поэтому в дальнейшем особь и хромосома идентичные понятия.

Расстояние — хеммингово расстояние между бинарными хромосомами. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Кроссинговер (кроссовер) — операция, при которой две хромосомы обмениваются своими частями. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Мутация — случайное изменение одной или нескольких позиций в хромосоме. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Инверсия — изменение порядка следования битов в хромосоме или в ее фрагменте. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Популяция — совокупность индивидуумов.

(См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Пригодность (приспособленность) — критерий или функция, экстремум которой следует найти. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Лocus — позиция гена в хромосоме. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

Аллель — совокупность подряд идущих генов.

(См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 12).

При турнирном отборе (tournament selection) из популяции, содержащей  $N$  особей, выбираются случайным образом  $t$  особей, и лучшая из них особь записывается в промежуточный массив. Эта операция повторяется  $N$  раз. Особи в полученном промежуточном массиве затем используются для скрещивания (также случайным образом). Размер группы строк, отбираемых для турнира, часто равен 2. В этом случае говорят о двоичном (парном) турнире. Вообще же  $t$  называют численностью турнира.

(См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 16).

В методе рулетки (roulette-wheel selection) особи отбираются с помощью  $N$  «запусков» рулетки, где  $N$  — размер популяции. Колесо рулетки содержит по одному сектору для каждого члена популяции. При таком отборе члены популяции с более высокой приспособленностью с большей вероятностью будут чаще выбираться, чем особи с низкой приспособленностью

(См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 18).

Рекомбинация (вопроизведение)

Оператор рекомбинации применяют сразу же после оператора отбора родителей для получения новых особей-потомков. Смысл рекомбинации заключается в том, что

созданные потомки должны наследовать генную информацию от обоих родителей. Различают дискретную рекомбинацию и кроссинговер. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 18).

Дискретная рекомбинация (Discrete recombination) в основном применяется к хромосомам с вещественными генами. Основными способами дискретной рекомбинации являются собственно дискретная рекомбинация, промежуточная, линейная и расширенно линейная рекомбинации. Дискретная рекомбинация соответствует обмену генами между особями. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 20).

Промежуточная рекомбинация (Intermediate recombination) применима только к вещественным переменным, но не к бинарным. В данном методе предварительно определяется числовой интервал значений генов потомков, который должен содержать значения генов родителей. При промежуточной рекомбинации возникают значения генов, отличные от значения генов особей-родителей. Это приводит к возникновению новых особей, пригодность которых может быть лучше, чем пригодность родителей. В литературе такой оператор рекомбинации иногда называется дифференциальным скрещиванием.

(См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 20).

Линейная рекомбинация (Line recombination) отличается от промежуточной тем, что множитель  $\alpha$  выбирается для каждого потомка один раз.

Расстояние — хеммингово расстояние между бинарными хромосомами. Кроссинговер (кроссовер) — операция, при которой две хромосомы обмениваются своими частями.

Мутация — случайное изменение одной или нескольких позиций в хромосоме.

Инверсия — изменение порядка следования битов в хромосоме или в ее фрагменте.

Популяция — совокупность индивидуумов.

Пригодность (приспособленность) — критерий или функция, экстремум которой следует найти.

Локус — позиция гена в хромосоме

Аллель — совокупность подряд идущих генов.

Эпистаз — влияние гена на пригодность индивидуума в зависимости от значения гена, присутствующего в другом месте

В двухточечном кроссинговере (и многоточечном кроссинговере вообще) хромосомы рассматриваются как циклы, которые формируются соединением концов линейной хромосомы вместе. Для замены сегмента одного цикла сегментом другого цикла требуется выбор двух точек разреза. В

этом представлении, одноточечный кроссинговер может быть рассмотрен как кроссинговер с двумя точками, но с одной точкой разреза, зафиксированной в начале строки. Следовательно, двухточечный кроссинговер решает ту же самую задачу, что и одноточечный, но более полно. Хромосома, рассматриваемая как цикл, может содержать большее количество стандартных блоков, так как они могут совершить «циклический возврат». (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 30).

Однородный кроссинговер (Uniform crossover) создает маску (схему) особи, в каждом локусе которой находится потенциальная точка кроссинговера. Маска кроссинговера имеет ту же длину, что и скрещивающиеся особи. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 30).

Триадный кроссинговер (Triadic crossover). Данная разновидность кроссинговера отличается от однородного тем, что после отбора пары родителей из остальных членов популяции случайным образом выбирается особь, которая в дальнейшем используется в качестве маски. Далее 10 % генов маски мутируют. Затем гены первого родителя сравниваются с генами маски: если гены одинаковы, то они передаются первому потомку, в противном случае на соответствующие позиции хромосомы потомка переходят гены второго родителя. Генотип второго потомка отличается от генотипа первого тем, что на тех позициях, где у первого потомка стоят гены первого родителя, у второго потомка стоят гены второго родителя и наоборот. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 36).

Перетасовочный кроссинговер (Shuffler crossover). В данном алгоритме особи, отобранные для кроссинговера, случайным образом обмениваются генами. Затем выбирают точку для одноточечного кроссинговера и проводят обмен частями хромосом. После скрещивания созданные потомки вновь тасуются. Таким образом, при каждом кроссинговере создаются не только новые потомки, но и модифицируются родители (старые родители удаляются), что позволяет сократить число операций по сравнению с однородным кроссинговером. См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. С. 38).

Кроссинговер с уменьшением замены (Crossover with reduced surrogate). Оператор уменьшения замены ограничивает кроссинговер, чтобы всегда, когда это возможно, создавать новые особи. Это осуществляется за счет ограничения на выбор точки разреза: точки разреза должны появляться только там, где гены различаются. Как было показано выше, кроссинговер генерирует новое решение (в виде особи-потомка) на основе двух имеющихся, комбинируя их части. Поэтому число различных решений, которые могут быть получены кроссинговером при использовании одной и той же пары готовых решений, ограничено. (См: Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. 40).

## Генетические алгоритмы

Методы оптимизации комбинаторных задач различной степени сложности. Генетические алгоритмы. Базовый генетический алгоритм. Последовательные модификации базового генетического алгоритма. Параллельные модификации базового генетического алгоритма. Классификация генетических алгоритмов.

### Методы оптимизации комбинаторных задач различной степени сложности

В общем случае оптимизация или поиск наилучшего значения (набора параметров) некоторой заданной целевой функции является достаточно сложной задачей. Сложность оптимизации обуславливается, прежде всего, видом целевой функции, которая может иметь как глобальный, так и локальный оптимумы.

В настоящее время не существует метода оптимизации, который позволил бы решить любую задачу (был универсальным) и при этом однозначно определен как лучший среди других методов по точности решения.

По степени приближения к точному решению, а также по характеру пространства поиска задачи могут быть разделены на следующие категории.

**Комбинаторные задачи** - характеризуются конечным и дискретным пространством поиска. Сущность любой комбинаторной задачи можно сформулировать следующим образом: найти на множестве  $X$  элемент  $x$ , удовлетворяющий совокупности условий  $K(x)$ , в предположении, что пространство поиска  $X$  содержит некоторое конечное число различных точек.

**Общие задачи без ограничений** - имеют нелинейное и неограниченное пространство поиска. Методы оптимизации для таких задач обычно полагаются на правильность аналитической формулировки целевой функции. Оптимизация функции без ограничений заключается в максимизации или минимизации некоторой функции  $U(x_1, \dots, x_p)$ .

**Общие задачи с ограничениями** - могут быть сформулированы как задачи минимизации функции  $U(x_1, \dots, x_p)$  при следующих ограничениях:

$$g_i(x_1, \dots, x_p) \geq 0 \text{ для } 1 \leq i \leq m, h_j(x_1, \dots, x_p) = 0 \text{ для } 1 \leq j \leq n.$$

Обычно задачи с ограничениями могут быть сведены к задачам без ограничений с помощью метода штрафов.

Если пространство поиска содержит конечное число точек, то наиболее точное решение может быть уверенно получено методом полного перебора. Этот метод имеет один очевидный недостаток - сложность вычислений, а следовательно, время, затрачиваемое на нахождение оптимального решения, существенно зависит от размерности пространства поиска. Метод перебора может быть достаточно эффективным только в небольшом пространстве поиска.

Градиентные методы, являющиеся основой линейного и нелинейного, динамического программирования, а также численного анализа, более универсальны, но менее точны. При этом усложнение ландшафта пространства поиска приводит к снижению эффективности таких методов. Методы градиента не гарантируют получение единственного оптимального решения, за исключением случая, когда пространство отображения является выпуклым и не допускает появления второстепенных вершин, плато и т. д.

С другой стороны, **эвристические методы**, к которым относятся **генетические алгоритмы (ГА)**, являются наиболее универсальными, поэтому не гарантируют нахождения глобального оптимума, являющегося единственным решением задачи.

Характеристикой задачи и, соответственно, основой для классификации методов оптимизации является также сложность задачи. По степени сложности однозначно выделяются следующие задачи.

**Линейные задачи** - сложность которых определяется как  $O(n)$ , где  $n$  — размерность входных данных задачи.

**Полиномиальные задачи (P)** - для них известен алгоритм, сложность которого составляет полином заданной, постоянной и не зависящей от размерности входной величины  $n$  степени.

**Экспоненциальные задачи** - сложность которых не менее порядка  $f^n$ , где  $f$  - константа или полином от  $n$ .

Однако существует большое число задач, которые не попадают ни в один из перечисленных классов. Сложность решения таких задач не может быть определена априорно. К ним относятся: оптимизация пути коммивояжера, оптимальная загрузка емкости, оптимизация маршрутов, инвестиций и т. д.

В общем случае задача оптимизации в настоящее время не может быть отнесена к какому-либо классу.

ГА являются стохастическим эвристическим методом, в котором вероятность выбора состояния  $S(t+1)$  зависит от состояния  $S(t)$  и косвенно от предыдущих состояний. Стохастические методы позволяют решать широкий класс таких задач, поскольку не требуют жесткой формализации. Следует отметить, что стохастические методы оптимизации используются для решения NP-сложных комбинаторных задач, т. е. таких задач, к которым сводима любая задача из класса NP. При этом **NP-сложные задачи** не обязательно относятся к классу NP.

Каждый из стохастических и эвристических методов имеет свои достоинства и недостатки, обусловленные формулировкой и размерностью решаемой задачи. **При этом**

**эффективность всех алгоритмов для всех возможных задач одинакова.** На рис. 1 приведена классификация эвристических и стохастических алгоритмов.

Полученные в результате решения большого количества задач результаты, усредненные по 10 запускам, доказывают справедливость утверждения о сравнимости эффективности всех перечисленных алгоритмов поиска глобального оптимума. Вместе с тем результаты, полученные при одном запуске, говорят о наибольшей эффективности двух методов поиска - ГА и поиска с учетом запретов.

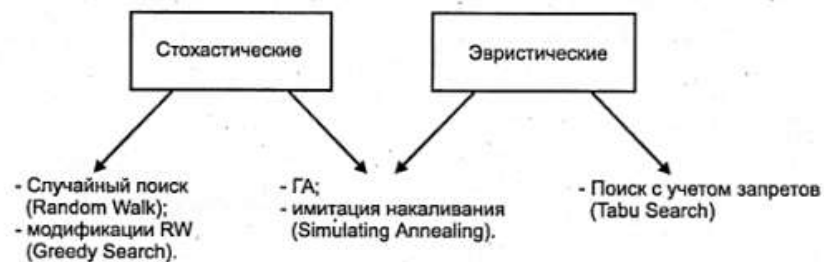


Рис. 1. Классификация эвристических и стохастических алгоритмов

Стохастические методы позволяют решать широкий класс таких задач, поскольку не требуют жесткой формализации.

Несмотря на некоторые различия в классификациях, ГА и поиск с учетом запретов имеют общую основу. Их объединяет использование эвристики для перехода из текущего состояния в последующее. **Особенностью ГА является работа с пространством поиска с помощью комбинирования решений, а поиска с учетом запретов - использование памяти состояний.**

Эффективностью обоих методов обусловлено появление метода НГТ (гибридной стратегии), использующей поиск с учетом запретов для повышения эффективности генетических операторов рекомбинации и мутации. Вместе с тем, если по эффективности отмеченные методы сравнимы, то по надежности поиск с учетом запретов уступает ГА, поскольку для данного метода качество решения существенно зависит от начального состояния (или решения). Поэтому начальное решение с высоким значением оценочной функции (в случае решения задачи минимизации) может быстро привести к желаемому решению, а начальное решение с низким значением оценочной функции - существенно снизить скорость поиска.

Анализ результатов использования ГА позволяет выделить следующие условия, при выполнении которых задача решается эффективно:

- **Q большое пространство поиска**, ландшафт которого является негладким (содержит несколько экстремумов);
- **Q сложность формализации оценки качества решения** функцией степени пригодности;
- **Q многокритериальность поиска;**

Генетические алгоритмы (ГА) относятся к числу универсальных методов оптимизации, позволяющих решать задачи различных типов (комбинаторные, общие задачи с ограничениями и без ограничений) и различной степени сложности. При этом ГА характеризуются возможностью как однокритериального, так и многокритериального поиска в большом пространстве, ландшафт которого является негладким.

В последние годы резко возросло число работ, прежде всего зарубежных ученых, посвященных развитию теории ГА и вопросам их практического использования. Результаты данных исследований показывают, в частности, что ГА могут получить более широкое распространение при интеграции с другими методами и технологиями. Появились работы, в которых доказывалась эффективность интеграции ГА и методов теории нечеткости, а также нейронных вычислений и систем.

Эффективность такой интеграции нашла практическое подтверждение в разработке соответствующих инструментальных средств (ИС). Так, фирма Attar Software включила ГА-компонент, ориентированный на решение задач оптимизации, в свои ИС, предназначенные для разработки экспертной системы. Фирма California Scientific Software связала ИС для нейронных сетей с ГА-компонентами, обеспечивающими автоматическую генерацию и настройку нейронной сети. Фирма NIBS Inc. включила в свои ИС для нейронных сетей, ориентированные на прогнозирование рынка ценных бумаг, ГА-компоненты, которые, по мнению финансовых экспертов, позволяют уточнять прогнозирование.

Несмотря на известные общие подходы к такой интеграции ГА и нечеткой логики, по-прежнему актуальна задача определения наиболее значимых параметров операционного базиса ГА с целью их адаптации в процессе работы ГА за счет использования нечеткого производственного алгоритма (НПА).

Перечисленные далее причины коммерческого успеха инструментальных средств в области искусственного интеллекта могут рассматриваться как общие требования к разработке систем анализа данных, используемых ГА:

**интегрированность** - разработка ИС, легко интегрирующихся с другими информационными технологиями и средствами;

**открытость и переносимость** - разработка ИС в соответствии со стандартами, обеспечивающими возможность исполнения в разнородном программно-аппаратном окружении, и переносимость на другие платформы без перепрограммирования;

**использование языков традиционного программирования** - переход к ИС, реализованным на языках традиционного программирования (С, С++ и т. д.), что упрощает

обеспечение интегрированности, снижает требования приложений к быстродействию ЭВМ и к объемам оперативной памяти;

**архитектура "клиент-сервер"** - разработка ИС, поддерживающих распределенные вычисления в архитектуре "клиент-сервер", что позволяет снизить стоимость оборудования, используемого в приложениях, децентрализовать приложения и повысить их производительность.

Перечисленные требования обусловлены необходимостью создания интегрированных приложений, т. е. приложений, объединяющих в рамках единого комплекса традиционные программные системы с системами искусственного интеллекта и ГА в частности.

**Интеграция ГА и нейронных сетей** позволяет решать проблемы поиска оптимальных значений весов входов нейронов, а **интеграция ГА и нечеткой логики** позволяет оптимизировать систему продукционных правил, которые могут быть использованы для управления операторами ГА (двунаправленная интеграция).

Одним из наиболее востребованных приложений ГА в области Data Mining является поиск наиболее оптимальной модели (поиск алгоритма, соответствующего специфике конкретной области).

### **Базовый генетический алгоритм**

Эволюционные алгоритмы, моделирующие процессы естественной эволюции, были предложены уже в 60-х годах прошлого века. Их особенностью является то, что они опираются на естественную эволюцию в природе, используя основные ее механизмы (отбор или селекцию, скрещивание и мутацию). Известны утверждения: "алгоритм является хорошим оптимизационным методом, потому что его принцип используется в природе", и наоборот: "алгоритм не может быть хорошим оптимизационным методом, потому что вы не находите его в природе".

Моделирование процесса естественной эволюции для эффективной оптимизации является первостепенной задачей теоретических и практических исследований в области эволюционных алгоритмов.

В 70-х годах прошлого века независимо друг от друга появились **два различных направления в области эволюционных алгоритмов: генетический алгоритм Холланда и эволюционные стратегии (ЭС) Речёнберга и Швифела**. Эволюционные стратегии используют операторы селекции и мутации, а если использовать биологические термины, то эволюционная стратегия моделирует естественную эволюцию с помощью непарной репродукции (рис. 2).

**Эволюционные стратегии ( $\mu + \lambda$ ).**

Шаг 1. Создание первоначальной популяции размера  $\lambda$ .

Шаг 2. Вычисление пригодности  $F(x_i)$   $i = 1, \dots, \lambda$ .

Шаг 3. Селекция (отбор)  $\mu < \lambda$  лучших индивидов.

Шаг 4. Создание  $\lambda / \mu$  потомков каждого из  $\mu$  индивидов с небольшими вариациями.

Шаг 5. Возврат к шагу 2.

Рис. 2. Разновидность эволюционных алгоритмов - эволюционные стратегии

Алгоритмы поиска, которые моделируют **парную репродукцию**, называются генетическими алгоритмами. Парная репродукция характеризуется рекомбинацией двух родительских строк для создания потомков. Эта рекомбинация называется **скрещиванием**.

**Предпочтение разных генетических операторов в ЭС и ГА** определило отношение к используемому размеру популяции. Так, Холланд подчеркивал важность рекомбинации в больших популяциях, в то время как Реченберг и Швэфел, главным образом, рассматривали мутацию в очень маленьких популяциях.

При работе с ГА решения задачи должны быть представлены в виде строки с бинарной, целочисленной или вещественной кодировкой. Способ кодирования предполагает работу со строками фиксированной или переменной длины, возможна также и контекстно-зависимая кодировка. Основным отличием генетических программ (ГП) от ГА является работа с деревьями решений. При этом в ГП отсутствует необходимость в генетическом представлении задачи. Такая схема представления вносит гибкость в описание структур данных, однако решения могут стать очень объемными без улучшения производительности. Это справедливо и для эволюционных программ (ЭП).

На рис. 3 приведен базовый или стандартный ГА (СГА), предложенный Холландом, который явился основой для различных модификаций.

**СГА.**

Шаг 0. Определение генетического представления задачи.

Шаг 1. Создание первоначальной популяции индивидов  $P(0) = x_1^0, \dots, x_N^0, t = 0$ .

Шаг 2. Вычисление средней пригодности  $f_{cp}(t) = \sum_i^N f(x_i) / N$ . Вычисление нормализованного значения степени пригодности  $f(x_i) / f_{cp}(t)$  для каждого индивида.

Шаг 3. Назначение каждому индивиду  $x_i$  вероятности  $p(x_i, t)$  пропорционально нормализованной пригодности. Выбор  $N$  векторов из  $P(t)$ , используя полученное распределение. Это дает набор отобранных родителей.

Шаг 4. Формирование случайным образом из данного набора  $N/2$  пар. Применение к каждой паре скрещивания, а также других генетических операторов, таких как мутация, для формирования новой популяции  $P(t + 1)$ .

Шаг 5.  $t = t + 1$ , возврат к шагу 2.

Рис. 3. Стандартный генетический алгоритм

## Последовательные модификации базового генетического алгоритма

Как показывает анализ, модификации ГА отличаются, прежде всего, способом селекции индивидов. В основных модификациях ГА несколько способов селекции используется для достижения различных целей - упрощения формирования промежуточной популяции, распараллеливания работы алгоритма, возможности анализа и предсказания поведения ГА. Было произведено сравнение четырех различных схем селекции (для CGA и SSGA, рассматриваемых далее, показавшее, что эффективность всех методов примерно одинакова. Таким образом, в настоящее время абсолютно лучший метод селекции не определен.

Модификация стандартного варианта ГА (Steady State GA) [Whitley и Kauth, 1988] затронула способ формирования промежуточной популяции (Mating Pool), являющейся результатом отбора (селекции) для формирования наследников с помощью генетических операторов. SSGA не формирует промежуточную популяцию как стандартный ГА, а осуществляют последовательно выбор пары наилучших индивидов, применяя к ним генетические операторы с целью формирования наследников, которые заменяют худшие индивиды популяции. Данная модификация ГА представлена на рис. 4.

**SSGA.**  
Шаг 0. Определение генетического представления задачи.  
Шаг 1. Создание первоначальной популяции  $P(0) = x_1^0, \dots, x_N^0, t = 0$ .  
Шаг 2. Вычисление относительной (нормализованной) степени пригодности  
 $f_r(x_i) = f(x_i) / \sum_i^N f(x_i) / N$ .  
Шаг 3. Выбор пары из лучших индивидов. Выбор худшего индивида. Применение скрещивания и мутации к выбранной паре лучших индивидов. Результат замещает худший индивид.  
Шаг 4.  $t = t + 1$ , возврат к шагу 2.

Рис. 4. Steady State GA

При проектировании ГА могут быть выгодно использованы знания, полученные селекционерами в области искусственной селекции. Генетические алгоритмы селекционеров (ГАС) моделируют именно искусственную селекцию. ГАС представлен на рис. 5, где под виртуальным селекционером понимается некоторый механизм селекции, который и является основным отличием ГАС от стандартного ГА.

**ГАС.**

Шаг 0. Определение генетического представления задачи.

Шаг 1. Создание первоначальной популяции  $P(0)$  размером  $N$ ,  $t = 0$ .

Шаг 2. Виртуальный селекционер отбирает  $T\%$  популяции для создания потомков. Это дает набор отобранных родителей.

Шаг 3. Формирование случайным образом из данного набора  $N/2$  пар. Применение к каждой паре скрещивания и мутации, формируя новую популяцию  $P(t + 1)$ .

Шаг 5.  $t = t + 1$ , возврат к шагу 2.

Шаг 6. Возврат к шагу 3.

Рис. 5. Генетический алгоритм селекционеров

Селекция основывается преимущественно на статистических методах, которые позволяют выполнить теоретический анализ и прогнозировать эффективность механизмов селекции, мутации и рекомбинации с помощью введенных уравнений селекции, реакции на селекцию и понятия наследственности.

Еще одна модификация ГА затрагивает решение многокритериальных задач. Многокритериальный ГА (МГА) также является модификацией стандартного ГА и отличается способом селекции, поскольку при отборе пар родителей в этом случае используется не один, а несколько критериев. При этом предлагается большое число вариантов схем селекции и соответственно вариантов МГА. На рис. 6 приведен вариант МГА, предложенный Schaffer в 1984 г., - векторный ГА (VEGA). Сравнительные оценки показывают, что по эффективности VEGA имеет средние показатели, однако не оценивалась вычислительная сложность для различных вариантов МГА, по которой VEGA может существенно улучшить свои показатели.

**Многокритериальный ГА.**

Шаг 0. Определение генетического представления задачи.

Шаг 1. Создание первоначальной популяции  $P(0) = x^0_1, \dots, x^0_N$ ,  $t = 0$ .

Шаг 2. Последовательное выполнение шагов 2.1–2.3.

Шаг 2.1. Вычисление значения степени пригодности каждого индивида по критерию  $i = 1, \dots, k$ .

Шаг 2.2. Для  $j$  от 1 до  $N/k$  осуществление селекции индивида из популяции в промежуточную популяцию.

Шаг 2.3. Возврат к шагу 2.1, если  $l < k$ .

Шаг 3. Формирование случайным образом из данного набора  $N/2$  пар. Применение к каждой паре скрещивания, а также других генетических операторов, таких как мутация, формируя новую популяцию  $P(t + 1)$ .

Шаг 4.  $t = t + 1$ , возврат к шагу 2.

Рис. 6. Многокритериальный генетический алгоритм

Стандартный ГА представляет собой строго синхронизованный последовательный алгоритм, который в условиях большого пространства поиска или сложного ландшафта пространства поиска может быть неэффективен по критерию времени. Эту проблему позволяет решить другой вид ГА - параллельный генетический алгоритм (ПГА). Следует отметить, что любая последовательная модификация стандартного ГА может быть преобразована в параллельную.

**По степени распараллеливания можно выделить следующие типы параллельных ГА:**

- ПГА на базе популяции;
- ПГА на базе подпопуляций;
- ПГА на базе индивидов.

**ПГА на базе популяции** сохраняет стандартную структуру ГА, работающего с целой популяцией, распараллеливание реализуется на этапе скрещивания и мутации (см. шаг 4, рис. 3). По степени распараллеливания процессов можно выделить следующие модели:

**синхронная модель "ведущий-ведомый"**, где главный процесс хранит целую популяцию в собственной памяти, выполняет селекцию, скрещивание и мутацию, но оставляет вычисление степени пригодности новых индивидов к подчиненным процессам;

**полусинхронная модель "ведущий-ведомый"**, где новый индивид обрабатывается по мере освобождения одного из процессов;

**асинхронная параллельная модель**, где индивиды популяции хранятся в общей памяти, к которой можно обращаться к параллельным процессам. Каждый процесс выполняет оценку степени пригодности, а также генетические операции.

Каждый процесс работает независимо от других. Единственное отличие между этой моделью и стандартным ГА заключается в механизме селекции. Очевидным в этом случае является вариант использования  $N/2$  параллельных процессоров при популяции в  $N$  индивидов. Тогда каждый процессор дважды случайным образом выбирает два индивида из общей памяти и оставляет лучшего. Два выбранных индивида затем подвергаются скрещиванию, мутации и оценке степени пригодности. Возникающие в результате наследники размещаются в общей памяти.

**Распределенный ПГА.**

Шаг 0. Определение генетического представления задачи.

Шаг 1. Создание первоначальной популяции индивидов и разделение на подпопуляции  $SP_1, \dots, SP_N$ .

Шаг 2. Формирование структуры подпопуляций.

Шаг 3. Для  $SP_i, i = 1, \dots, N$  — выполнение параллельно шагов 3.1–3.3.

Шаг 3.1. Применение в течение  $m$  поколений селекции и генетических операторов.

Шаг 3.2. Перемещение  $k$  хромосом в соседние подпопуляции.

Шаг 3.3. Получение хромосом из соседних подпопуляций.

Шаг 4. Возврат к шагу 3.

Рис. 7. Распределенный параллельный генетический алгоритм

Особенность **ПГА на базе подпопуляции** заключается в использовании независимых конкурирующих подпопуляций, которые обмениваются индивидами с заданной частотой (распределенный ПГА, рис. 7). При этом каждый процессорный блок выполняет последовательный ГА с собственной подпопуляцией, при условии максимизации одной общей для всех функции степени пригодности. В этом случае для обмена индивидами должна быть определена структура связей подпопуляций. С точки зрения оценки и сравнения эффективности может быть рассмотрен вариант распределенной модели, в которой обмен индивидами не осуществляется. Результаты проведенных экспериментов свидетельствуют о большей эффективности распределенного ПГА по сравнению с этим частным случаем, а также со стандартным ГА.

Существенным недостатком модели может стать снижение степени разнообразия при интенсивном обмене индивидами. С другой стороны, недостаточно частое перемещение может привести к преждевременной сходимости подпопуляций. При построении такой модели важно определить следующее:

связи между процессорами для обмена индивидами;

частоту обмена индивидами (оптимальной является частота обмена через 20 поколений);

степень перемещения или число обмениваемых индивидов (оптимальным является 20 % подпопуляции);

способ селекции индивида для обмена;

критерий, по которому полученный индивид сможет заменить члена подпопуляции,

С точки зрения времени и даже числа поколений, затрачиваемых на решение задачи, ПГА эффективнее стандартного ГА, но при этом некоторые задачи могут быть слишком простыми для ПГА. Параллельный поиск имеет смысл в том случае, если пространство поиска

большое и сложное. Увеличение числа процессоров в данной модели улучшает скорость сходимости, но не качество решения.

**ПГА на базе индивидов** имеют одну строку индивида, постоянно находящуюся в каждом процессорном элементе (ячейке). Индивиды выбирают пары и рекомбинируют с другими индивидами в их непосредственном ближайшем окружении (по вертикали и горизонтали). Выбранный индивид затем совмещается с индивидом, постоянно находящимся в ячейке. В результате формируется один наследник, который может или не может заменить индивида в ячейке в зависимости от выбранной схемы замещения. Таким образом, модель является полностью распределенной и не нуждается в централизованном управлении (рис. 8).

**ПГА на базе индивидов.**  
Шаг 0. Определение генетического представления задачи.  
Шаг 1. Создание первоначальной популяции индивидов и формирование структуры популяции.  
Шаг 2. Локальное повышение каждым индивидом своей производительности (hill-climbing).  
Шаг 3. Выполнение каждым индивидом селекции с целью поиска пары.  
Шаг 4. Применение к паре скрещивания, а также других генетических операторов, таких как мутация.  
Шаг 5. Локальное повышение наследником своей производительности (hill-climbing).  
Замещение наследником родителя в соответствии с заданным критерием качества.  
Шаг 6. Возврат к шагу 3.

Рис. 8. Параллельный генетический алгоритм на базе индивидов

При работе с моделью на базе индивидов необходимо задать:

структуру связей ячеек;

схему селекции;

схему замещения.

Исследования этой модели показали, что для сложных задач она способна обеспечить лучшие решения, чем стандартный ГА.

### **Классификация генетических алгоритмов**

В ходе исследований в области генетических алгоритмов и эволюционных алгоритмов в целом появилось большое количество направлений, и их число непрерывно растет.

Классификация ЭА и основные модификации стандартного ГА, приведенного на рис. 2, отражены на рис. 9.

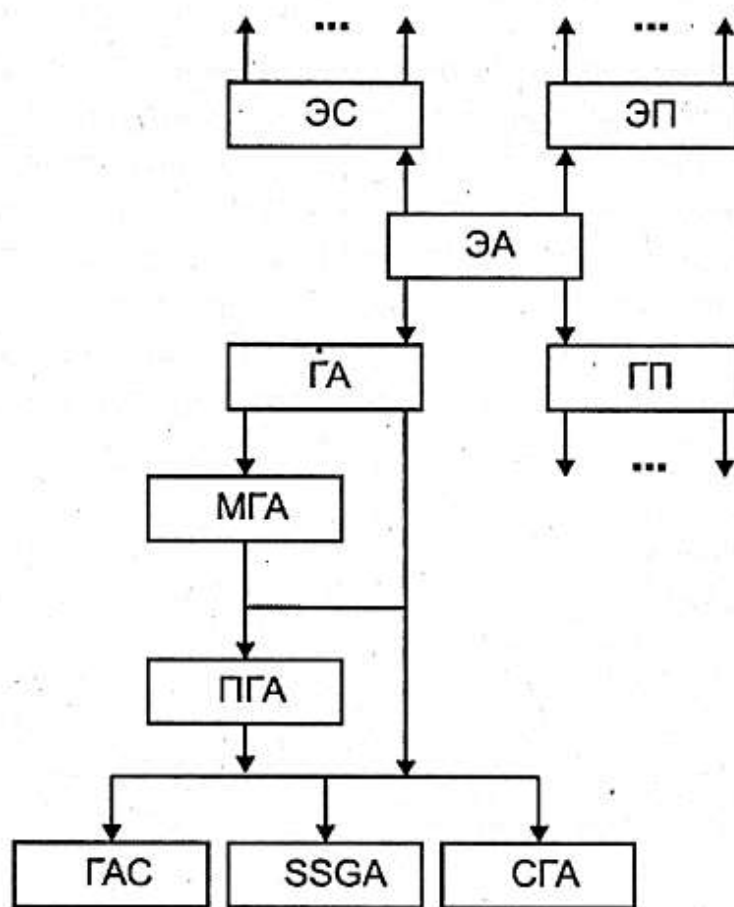


Рис. 9. Классификация эволюционных алгоритмов