

ИИС

03

ЭВОЛЮЦИОННЫЕ ВЫЧИСЛЕНИЯ



- * **эволюционные вычисления не гарантируют обнаружения глобального экстремума целевой функции (оптимального решения) за определенное время. Основное их преимущество в том, что они позволяют найти «хорошие» решения очень трудных задач за меньшее время, чем другие методы. Методы эволюционных вычислений оказались достаточно эффективными для решения ряда реальных задач инженерного проектирования, планирования, маршрутизации и размещения, управления портфелями ценных бумаг, прогнозирования, а также во многих других областях.**

- * **Системы символического обучения** ориентированы на добычу знаний (англ. Data-mining), поиск скрытых правил и закономерностей в компьютерных базах данных (англ. Knowledge Discovery), автоматические рассуждения, доказательство теорем и т.д. Для последних систем задача (проблема) и относящаяся к ней информация описывается в виде логических аксиом. В дальнейшем система рассматривает различные варианты задачи как теоремы, которые следует доказать.
- * В **нейросетевых системах**, построенных на принципах нервной системы биологических организмов, используются методы обучения, направленные на модификацию собственной структуры (структуры сети) и весовых коэффициентов связей между элементами.
- * **Эволюционные системы** построены на принципах генетических и эволюционных процессов в природе, когда из набора кандидатов (популяции), получаемого посредством скрещивания и мутаций, по принятому критерию отбираются лучшие, наиболее приспособленные для решения задачи.

Основные направления эволюционных вычислений

- * - эволюционное программирование;
- * - эволюционные стратегии;
- * - генетические алгоритмы.

Эволюционное программирование

- * автор Лоуренс Дж. Фогель, 1960 год)
- * идея представления альтернатив решения задачи в виде универсальных конечных автоматов, способных реагировать на стимулы, поступающие из окружающей среды.
- * применено к решению прикладных задач, включая маршрутизацию трафика и планирование, выявление рака, военное планирование, обучение в играх, разработку систем управления, идентификации, обработки сигналов

- * **Гипотезы** о виде зависимости целевой функции от переменных формулируются системой **в виде программ** на некотором внутреннем языке программирования.
- * Процесс построения таких программ строится как эволюция в мире программ.
- * Система находит программу, которая достаточно точно выражает искомую зависимость,
- * Начинает вносить в нее небольшие модификации
- * Отбирает среди построенных таким образом дочерних программ те, которые повышают точность.
- * Система «выращивает» несколько генетических линий программ, конкурирующих между собой в точности нахождения искомой зависимости.
- * Специальный транслирующий модуль, переводит найденные зависимости с внутреннего языка системы на понятный пользователю язык (математические формулы, таблицы и др.)

Эволюционные стратегия

- * автор Инго Рехенберг, 1964 год
- * каждая из альтернатив представляется вектором действительных чисел.
- * В качестве мутации часто используется добавление нормально распределенной случайной величины к каждой компоненте вектора.
- * параметры нормального распределения самоадаптируются в процессе выполнения алгоритма. Другой отличительной особенностью эволюционных стратегий является детерминированный отбор лучших особей из родителей и порожденных потомков без повторений

- * 1. система чувствительна ко времени
- * Min времени Max точность

генетические алгоритмы

- * автор Джон Холланд, 1975 год
- * представление любой альтернативы решения в виде кодовой (как правило, битовой) строки фиксированной длины,
- * манипуляции с которой производятся в отсутствие всякой связи с ее смысловой интерпретацией

Определения

- * **Ген (свойство)** – атомарный элемент хромосомы. Ген может быть битом, числом или неким другим объектом.
- * **Аппель** – значение конкретного гена.
- * **Локус** – положение конкретного гена в хромосоме.
- * **Хромосома (цепочка)** – упорядоченная последовательность генов.
- * **Генотип (код)** – упорядоченная последовательность хромосом.
- * **Особь (индивидуум)** – конкретный экземпляр генотипа.
- * **Фенотип** – аргумент (набор аргументов) целевой функции, соответствующий генотипу (т.е. интерпретация генотипа с точки зрения решаемой задачи).

- * **генотип** состоит из одной хромосомы и представляется в виде битовой строки.
- * Тогда ген – это бит;
- * генотип (хромосома) – битовая строка, заданной размерности и с определенным положением битов; особь – конкретный набор битов (0 и 1).

- 
- * На каждом шаге работы генетический алгоритм использует несколько точек поиска решения одновременно.
 - * Совокупность этих точек является набором особей, который называется **популяцией**.
 - * Количество особей в популяции называют **размером популяции**.

- 
- * На каждом шаге работы генетический алгоритм обновляет популяцию путем создания новых особей и уничтожения ненужных.
 - * Чтобы отличать популяции на каждом из шагов и сами эти шаги, их называют **поколениями** и обычно идентифицируют по номеру.
 - * популяция, полученная из исходной популяции после первого шага работы алгоритма, будет первым поколением, после следующего шага - вторым и т. д.

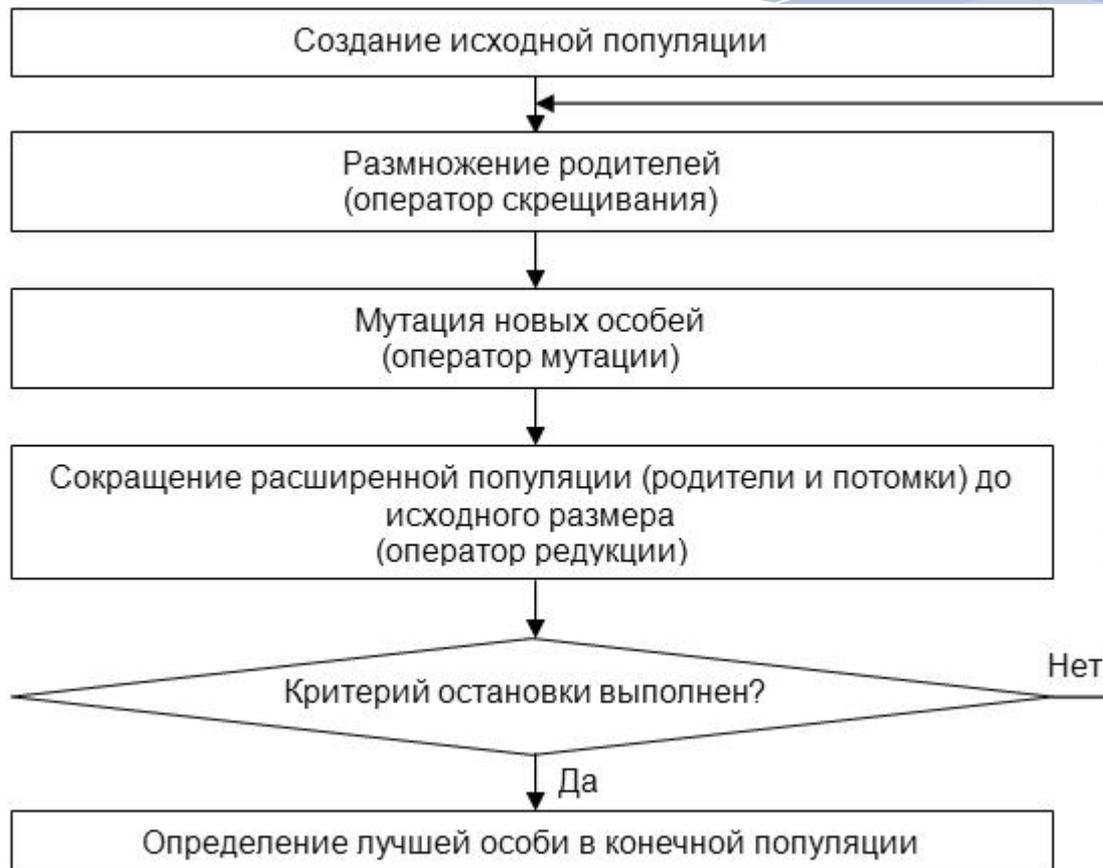
- * Генерация новых особей происходит на основе моделирования процесса **размножения** с помощью **оператора скрещивания**.
- * порождающие особи называются **родителями**,
- * порожденные - **потомками**.
- * Выбор родителей для скрещивания выполняется случайным образом.
- * При этом, один родитель может участвовать в нескольких скрещиваниях и необязательно, чтобы все родители участвовали в размножении.
- * выбор родителей выполняется случайным образом, то может получиться, что одна особь скрещивается сама с собой.
- * Родительская пара, как правило, порождает пару потомков. Непосредственная генерация новых кодовых строк из двух выбранных происходит за счет обмена между родителями случайно выбранными наборами генов

- * Моделирование процесса мутации новых особей осуществляется за счет работы **оператора мутации**, применяемого к случайно выбранным потомкам за счет изменения случайного выбранного гена.
- * размер популяции фиксирован, то порождение потомков должно сопровождаться уничтожением особей.
- * Выбор лучших («жизнеспособных») особей из числа родителей и потомков выполняется в **операторе редукции**, который уничтожает худшие («малоприспособленные») особи.
- * Основным правилом отбора является **закон эволюции: «выживает сильнейший»**, который обеспечивает улучшение искомого решения. В некоторых источниках процесс приведения расширенной популяции к исходному размеру рассматривается не с точки зрения уничтожения худших особей (редукции), а с точки зрения выбора лучших.
- * Операторы скрещивания, мутации и редукции называют **генетическими операторами**. Скрещивание и мутация выполняются с использованием элементов случайности, а редукция - по строго детерминированным правилам.

Основные отличия генетических алгоритмов от традиционных методов поиска решений

- * 1. Генетические алгоритмы работают с кодовыми строками, от которых зависят значения аргументов целевой функции и, соответственно, значение самой целевой функции. Интерпретация этих кодов выполняется только в операторе редукции. В остальном работа алгоритма не зависит от смысловой интерпретации кодов (фенотипа), т.е. никак не связана с сущностью (спецификой) решаемой задачи.
- * 2. Для поиска лучшего решения генетический алгоритм на отдельном шаге использует сразу несколько точек поискового пространства (несколько вариантов решения задачи) одновременно, а не переходит от точки к точке, как это делается в традиционных методах математического программирования. Это позволяет преодолеть один из их недостатков - опасность попадания в локальный экстремум целевой функции, если она не является унимодальной (т.е. имеет несколько экстремумов). Использование нескольких точек одновременно значительно снижает такую возможность.
- * 3. Генетический алгоритм использует как вероятностные правила для порождения новых точек для анализа, так и детерминированные для перехода от одних точек к другим

Последовательность работы генетического алгоритма



Критерий остановки работы генетического алгоритма

- * - сформировано заданное число поколений;
- * - исчерпано время, отведенное на эволюцию;
- * - популяция достигла заданного качества (значение критерия одной (нескольких, всех) особей превысило заданный порог);
- * - достигнут некоторый уровень сходимости (особи в популяции стали настолько подобными, что дальнейшее их улучшение происходит чрезвычайно медленно);

Пример работы генетического алгоритма при поиске максимума функции одной переменной

- * Пусть имеется набор натуральных чисел от 0 до 31 и функция, определенная на этом наборе чисел, $f(x) = x$. Требуется найти максимальное значение функции. Задача, конечно, тривиальна и не требует применения столь изощренных методов поиска, но мы решаем ее лишь для иллюстрации работы генетического алгоритма

- * В качестве кодовой строки использовать двоичное представление аргументов функции.
- * Таким образом, ген - это отдельный бит строки, хромосома - последовательность битов (для чисел от 0 до 31 длина кодовой строки - 5 бит), генотип состоит из одной хромосомы (генотип = хромосома), фенотип - десятичное представление кодовой (битовой) строки (он же и является значением целевой функции).
- * Определить некоторые характеристики генетического алгоритма. Пусть размер популяции будет 4 особи, число скрещиваний - 2, число мутаций - 1 на поколение. Процесс мутации заключается в инверсии одного из битов строки, выбирается случайно.

Пример

- 
- * случайным образом создана исходная популяция из четырех особей

№ строки	Код	Фенотип, x	Значение целевой функции, $f(x) = x$
1	01011	11	11
2	10010	18	18
3	00010	2	2
4	01100	12	12
Среднее значение			10.75

- 
- * оператор отбора выбрал для производства потомков две пары строк (1, 2) и (2, 4). Работа оператора скрещивания
 - * При этом в каждой паре разбиение на подстроки происходит независимо и случайным образом.
 - *

№ строки	Родители	Потомки
1	0 1011	00010
2	1 0010	11011
3	100 10	10000
4	011 00	01110

- 
- * Пусть оператор мутации сработал для младшего бита потомка в строке 3 и данный код изменился с 10000 на 10001.
 - * Таким образом, популяция за счет порожденных потомков расширилась до восьми особей

№ строки	Код	Фенотип, x	Значение целевой функции, $f(x) = x$
Исходная популяция			
1	01011	11	11
2	10010	18	18
3	00010	2	2
4	01100	12	12
Порожденные потомки			
5	00010	2	2
6	11011	27	27
7	10001	17	17
8	01110	14	14

- 
- * Оператор редукции далее сократит популяцию до исходного числа особей, исключив из нее те, значение целевой функции которых минимально.
 - * То есть будут исключены строки 1, 3, 4 и 5, и популяция первого поколения примет вид,

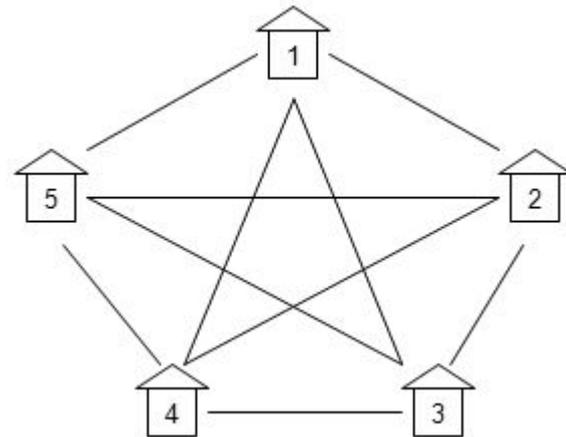
№ строки	Код	Фенотип, x	Значение целевой функции, $f(x) = x$
1	10010	18	18
2	11011	27	27
3	10001	17	17
4	01110	14	14
Среднее значение			19



* даже за одну итерацию качество популяции значительно возросло. Если в исходной популяции среднее значение целевой функции было 10.75, а ее минимальное значение составляло 2, то в популяции первого поколения среднее значение возросло до 19, а минимальное значение составило 14. Лучшее же решение увеличилось с 18 до 27 при оптимальном решении 31.

Пример работы генетического алгоритма при поиске решения задачи коммивояжера

- * **Постановка задачи** – коммивояжеру требуется посетить N городов. Для каждой пары городов по маршруту следования установлена стоимость (расстояние, время) проезда. Требуется найти путь минимальной стоимости, который начинается из некоторого города, обеспечивает посещение всех остальных городов ровно по одному разу и возврат в точку отправления.



*

Задача коммивояжера относится к категории NP-полных задач, т.е. задач решаемых методом полного перебора всех вариантов

Формализация задачи

- * Ген – число, характеризующее номер посещаемого города.
- * Хромосома – строка из чисел длиной N , характеризующая порядок посещения городов.
- * Генотип состоит из одной хромосомы.
- * Фенотип - порядок посещения городов (совпадает с генотипом).
- * Особь – конкретная строка из чисел (допустимый вариант решения задачи).
- * Предположим $N = 9$.
- * Особи «231586479» и «147523869» - примеры допустимых вариантов решения задачи.
- * Классическое **скрещивание** приведет к генерации недопустимых вариантов, например

Родители

23158 | 6479

14752 | 3869

Потомки

23158 | 3869

14752 | 6479

- * потомках посещение некоторых городов будет дублироваться или проигнорировано.
- * Рядом исследователей предложены различные варианты решения данной проблемы, в частности **Л. Девис** предлагает следующую модификацию оператора скрещивания.
- * 1) Случайным образом выбираются два сечения генотипа
- * $P_1 = 231 \mid 586 \mid 479$
- * $P_2 = 147 \mid 523 \mid 869$

- * 2) Для потомков копируются участки кода, расположенные между сечениями
- * $P_1 = xxx \mid 586 \mid xxx$
- * $P_2 = xxx \mid 523 \mid xxx$
- * 3) Из родителей генерируются вспомогательные строки, у которых участки кода циклически смещаются вправо.
- * $V_1 = 479 \ 231 \ 586$
- * $V_2 = 869 \ 147 \ 523$

- * 4) Свободные гены потомков последовательно заполняются генами из перекрестных вспомогательных строк с пропуском уже имеющихся в потомке генов
- * $P_1 = 914 \mid 586 \mid 723$
- * $P_2 = 479 \mid 523 \mid 186$
- * Оператор **мутации** также может быть реализован различными способами, например:
 - * 1) перестановка пары, случайным образом выбранных генов местами: $479523186 \rightarrow 473529186$;
 - * 2) инверсия случайным образом выбранной последовательности генов: $479 \mid 523 \mid 186 \rightarrow 479 \mid 325 \mid 186$.