Лекция № 9

Генетические алгоритмы

Genetic Algorithms

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Генетические алгоритмы (ГА) есть поисковые алгоритмы, основанные на механизмах натуральной селекции и натуральной генетики.
Основой для возникновения генетических алгоритмов считается **модель биологической эволюции** и **методы случайного поиска**.

Случайный поиск возник как реализация простейшей модели эволюции, когда случайные мутации моделировались случайными шагами оптимального решения, а отбор "уходом" неудачных вариантов.

Эволюционный поиск — это последовательное преобразование одного *конечного множества промежуточных решений* в другое. Само преобразование можно назвать алгоритмом поиска или алгоритмом эволюции.

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Замечания:

• каждое решение кодируется строкой символов (или битов), которую называют *стрингом* или *хромосомой:*

$$A_i = a_{i1}\, a_{i1} \ldots a_{in};$$

• при этом отдельный символ строки (или бит) a_{ij} кодирует какой-то параметр и называется геном:

•множество решений $\{A_1, A_2, A_m\}$, закодированных в виде хромосом, называют *популяцией*;

• результатом эволюционного поиска является не одно решение, а множество решений A_i.

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Для сравнения решений (хромосом, особей) между собой вводится “функция приспособленности” (fitness function) или **целевая функция**, которая показывает насколько хорошо данное решение соответствуют поставленной задаче.

Это функция от многих переменных (значений параметров, заданных в решении).

Как правило, чем больше значение целевой функции, тем лучше решение. Но в ряде задач требуется минимизировать целевую функцию.

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Выделяют **три особенности алгоритма эволюции**:

1) каждая новая популяция состоит только из "жизнеспособных" хромосом;
2) каждая новая популяция "лучше" (в смысле целевой функции) предыдущей;
3) в процессе эволюции последующая популяция зависит только от предыдущей.

Природа, реализуя эволюцию, как бы решает оптимизационную задачу на основе случайного поиска.

Генетический алгоритм — это алгоритм, который позволяет найти удовлетворительное решение к аналитически неразрешимым проблемам через последовательный подбор и комбинирование значений искомых параметров с использованием механизмов, напоминающих биологическую эволюцию.

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Простой генетический алгоритм был впервые описан В. Голдбергом на основе работ Дж. Холланда.

Предварительно ГА случайно генерирует начальную популяцию строк — стрингов (хромосом) вида:

$$a_1\, a_2 \ldots a_n ,$$

здесь a_i кодирует i-тый параметр решения, а n – число параметров.

Затем ГА применяет множество простых операций к начальной популяции и генерирует новые популяции.

Простой ГА состоит из трёх операторов:

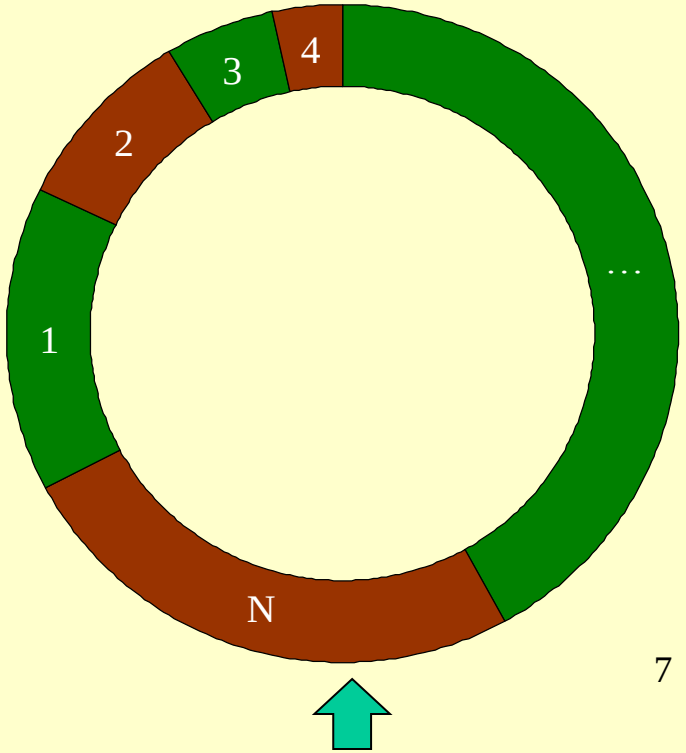
- *репродукция*;
- *кроссинговер*;
- *мутация*.

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Репродукция – процесс, в котором хромосомы копируются согласно их целевой функции (*ЦФ*) $f_i(x)$. Копирование хромосом с лучшим значением *ЦФ* имеет большую вероятность для их попадания в следующую генерацию.

Оператор репродукции (*ОР*) является искусственной версией натуральной селекции (“выживания сильнейших”) по Дарвину.

ОР в алгоритмической форме может представляться различными способами. Самый простой — создать колесо рулетки, в котором каждая хромосома имеет поле, пропорциональное его *ЦФ*.



ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Колесо рулетки вращается и после останова ее указатель определяет хромосому, выбранную для участия в следующем поколении.
При выполнении оператора ОР колесо рулетки вращается, такое количество раз, которое соответствует мощности начальной популяции.

При этом вероятность выбора *i-той* хромосомы вычисляется как:

$$p_i(OP) = f_i(x)/sum\ f_j(x),$$

$f_i(x)$ – *ЦФ i-той* хромосомы в популяции,

sum $f_j(x)$ – сумма *ЦФ* всех хромосом в популяции.

Ожидаемое число копий *i-той* хромосомы в новой популяции:

$$N = p_i(OP) \bullet n.$$

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Оператор кроссинговера (***ОК***) выполняется в 3 шага.
На первом шаге члены нового репродуцированного множества хромосом выбираются сначала.
Далее каждая пара хромосом (стрингов) пересекается по следующему правилу: целая позиция *k* вдоль стринга выбирается случайно между *1* и длиной хромосомы *L*, уменьшенной на единицу, т.е. в интервале *(1, L-1)*.

Следуя традициям генетики, хромосомы 1 и 2 часто называют родителями, а хромосомы 1', 2' — их потомками (детьми).
Число *k*, выбранное случайно, называется *точкой ОК* или *точкой разрыва ОК,* или *точкой пересечения ОК.*

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Итак, согласно Дж. Холланду *ОК* выполняется в три этапа:

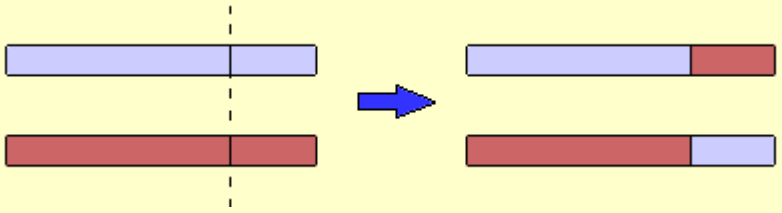
1. Две хромосомы $A = a_1\ a_2\ a_k\ a_{k+1}\ ...\ a_L$ и $B = b_1\ b_2\ b_k\ b_{k+1}\ ...\ b_L$ выбираются случайным образом из текущей популяции после *ОР*.

2. Число k выбирается из интервала $[1, L-1]$ также случайным образом. Здесь L длина хромосомы; k — точка ОК.

3. Две новые хромосомы формируются из A и B путем замены множества элементов. В результате получим две новые хромосомы:

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

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



Таким образом, механизмы *ОР* и *ОК*. включают случайную генерацию чисел, копирование хромосом и частичный обмен информацией между хромосомами.

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Далее согласно схеме классического ПГА, выполняется **оператор мутации (*ОМ*)**.

Согласно Дж. Холланду, *ОМ* состоит из двух этапов:

1. В хромосоме $A = a_1\ a_2\ a_3\ \ldots\ a_{m-1}\ a_m\ \ldots a_L$ определяются случайным образом две позиции, например позиция 2 и позиция m.

2. Гены, соответствующие выбранным позициям, меняются местами и формируется новая хромосома:

$$A' = a_1\ a_m\ a_3\ \ldots\ a_{m-1}\ a_2\ \ldots a_L$$

Следует заметить, что оператор мутации играет вторичную роль в ПГА. Поэтому обычно выбирают одну мутацию на 1000 циклов.

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Оператор мутации, необходим для *"выбивания" популяции из локального экстремума* и способствует *защите от преждевременной сходимости.*

Мутация необходима особенно для ГА с малым размером популяции, потому что для них свойственна преждевременная сходимость (premature convergence) – ситуация, когда в некоторых позициях все особи (хромосомы) имеют один и тот же бит, не соответствующий глобальному экстремуму.

Т.е. мутация нужна для того, чтобы внести новизну в множество решений, так как если во всех хромосомах в позиции *j* (3) стоит 0, то без *ОМ* там никогда не появится 1, так как при выполнении оператора *ОК* хромосомы только меняются своими частями, не изменяя конкретные гены.

A = 1001010010001

B = 0101101110001

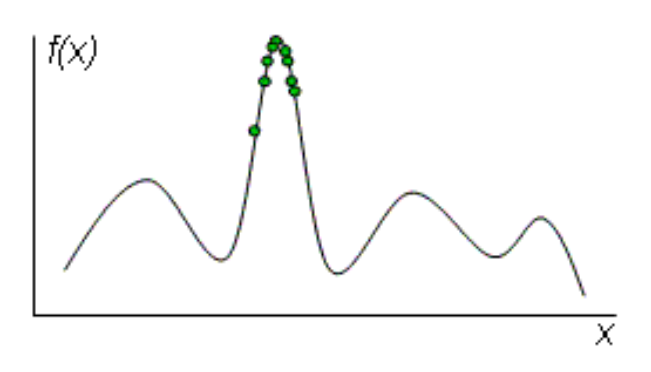
ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Критерием останова ГА может служить *(1) заданное количество поколений*, *(2) достижение определенного значения целевой функции* (у отдельного стринга) или *(3)* ***схождение*** *популяции*.

Схождением называется состояние популяции, когда все строки находятся в области некоторого экстремума и почти одинаковы.

Таким образом, схождение популяции означает, что достигнуто решение близкое к оптимальному.

Итоговым решением задачи может служить наиболее приспособленная особь (с наибольшим значением *ЦФ*) последнего поколения.



ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Замечания.

Следует заметить, что операторы *ОК* и *ОМ,* вообще говоря, 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 невозможно и необходимы другие механизмы поиска. Здесь как раз и приходит на помощь направленно-случайный поиск, который можно реализовать на основа ПГА.

Применение генетических алгоритмов

Генетические алгоритмы применяются в основном там, где сложно или невозможно сформулировать задачу в виде, пригодном для более быстрых алгоритмов локальной оптимизации, либо если стоит задача оптимизации недифференцируемой функции или задача многоэкстремальной глобальной оптимизации.
С помощью генетических алгоритмов можно находить квазиоптимальные решения чуть ли не 99% задач принятия решения.

Применение генетических алгоритмов

Применение генетических алгоритмов для оптимизации многопараметрических функций.

Большинство реальных задач могут быть сформулированы как поиск оптимального значения, где значение – сложная функция, зависящая от определенных входных параметров. В некоторых случаях, нужно найти те значения параметров, при которых достигается точное значение функции. В других случаях, точный оптимум не нужен – решением может считаться любое значение, превосходящее заданную величину. В этом случае, генетические алгоритмы – приемлемый метод для поиска "приемлемых" значений.

Применение генетических алгоритмов

- Оптимизация функций
- Оптимизация запросов в базах данных
- Разнообразные задачи на графах (задача коммивояжера, раскраска графа и т. п.) и комбинаторной оптимизации (например, задача о ранце)
- Задачи компоновки (в САПР)
- Составление расписаний
- Игровые стратегии
- Аппроксимация функций
- Искусственная жизнь (моделирование эволюции)
- Биоинформатика
- Настройка и обучение искусственной нейронной сети

Применение ГА 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, для которых еще не найден полиномиальный алгоритм решения.

Применение ГА при поиске решения задачи коммивояжера

Формализация задачи.

Ген – число, характеризующее номер посещаемого города.

Хромосома – строка из чисел длиной N, характеризующая порядок посещения городов.

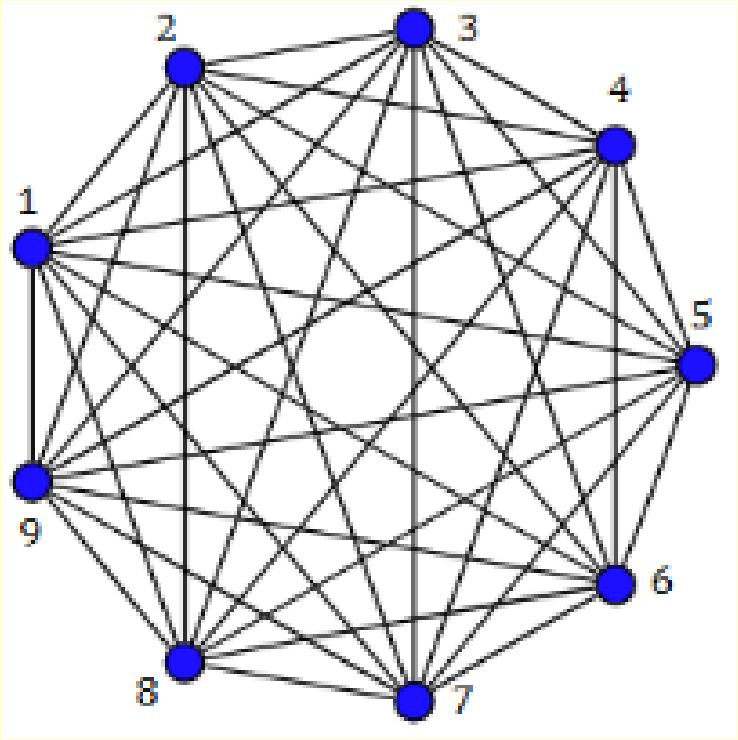
Генотип состоит из одной хромосомы.

Фенотип – порядок посещения городов (совпадает с генотипом).

Особь – конкретная строка из чисел (допустимый вариант решения задачи).

Применение ГА при поиске решения задачи коммивояжера

Предположим коммивояжеру необходимо посетить 9 городов, N = 9.



Особи «231586479» и «147523869» – примеры допустимых вариантов решения задачи.

Применение ГА при поиске решения задачи коммивояжера

Классическое скрещивание приведет к генерации недопустимых вариантов, например

Родители	Потомки
23158 | 6479	23158 | 3869
14752 | 3869	14752 | 6479

т.к. в потомках посещение некоторых городов будет дублироваться или проигнорировано.

Рядом исследователей предложены различные варианты решения данной проблемы.

Применение ГА при поиске решения задачи коммивояжера

Л. Девис предлагает следующую модификацию оператора скрещивания:

1) Случайным образом выбираются два сечения генотипа (хромосомы)
P_1 = 231 | 586 | 479
P_2 = 147 | 523 | 869

2) Для потомков копируются участки кода, расположенные между сечениями
$П_1$ = xxx | 586 | xxx
$П_2$ = xxx | 523 | xxx

3) Из родителей генерируются вспомогательные строки, у которых участки кода циклически смещаются вправо.
B_1 = 479 231 586
B_2 = 869 147 523

4) Свободные гены потомков (x) последовательно заполняются генами из перекрестных вспомогательных строк (см. п.3) с пропуском уже имеющихся в потомке генов
$П_1$ = 914 | 586 | 723
$П_2$ = 479 | 523 | 186

Применение ГА при поиске решения задачи коммивояжера

Оператор **мутации** также может быть реализован различными способами, например:

1) перестановка пары, случайным образом выбранных генов местами:

47**9**52**3**186 → 47**3**52**9**186;

2) инверсия случайным образом выбранной последовательности генов:

479 | **523** | 186 → 479 | **325** | 186.