

Лабораторная работа №6

Программирование генетического алгоритма

Цель работы: Создать на языке высокого уровня простейший генетический алгоритм, способный подстраиваться под условия среды.

Генетический алгоритм (англ. genetic algorithm) — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе. Является разновидностью эволюционных вычислений, с помощью которых решаются оптимизационные задачи с использованием методов естественной эволюции, таких как наследование, мутации, отбор и кроссинговер. Отличительной особенностью генетического алгоритма является акцент на использование оператора «скрещивания», который производит операцию рекомбинации решений-кандидатов, роль которой аналогична роли скрещивания в живой природе.

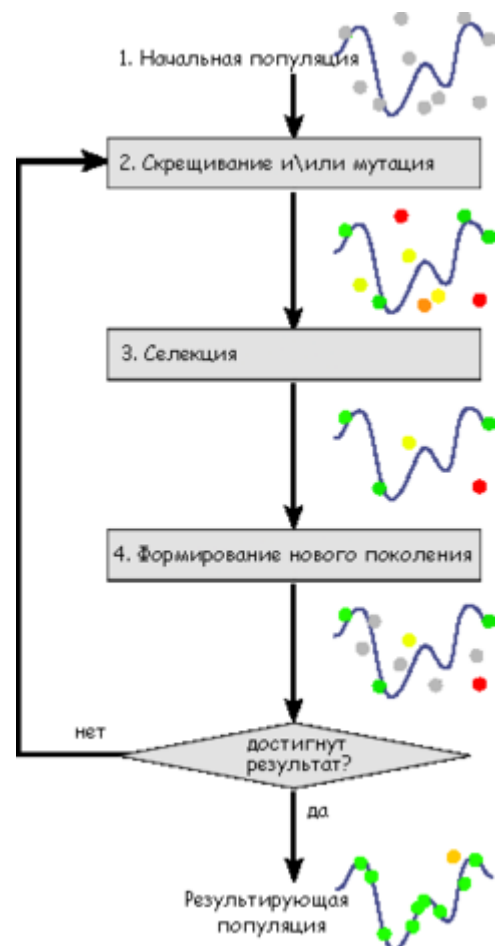
Схема работы генетического алгоритма

Задача формализуется таким образом, чтобы её решение могло быть закодировано в виде вектора («генотипа») генов, где каждый ген может быть битом, числом или неким другим объектом. В классических реализациях ГА предполагается, что генотип имеет фиксированную длину. Однако существуют вариации ГА, свободные от этого ограничения.

Некоторым, обычно случайным, образом создаётся множество генотипов начальной популяции. Они оцениваются с использованием «функции приспособленности», в результате чего с каждым генотипом ассоциируется определённое значение («приспособленность»), которое определяет насколько хорошо фенотип, им описываемый, решает поставленную задачу.

При выборе «функции приспособленности» (или fitness function в англоязычной литературе) важно следить, чтобы её «рельеф» был «гладким».

Из полученного множества решений («поколения») с учётом значения «приспособленности» выбираются решения (обычно лучшие особи имеют большую вероятность быть выбранными), к которым применяются «генетические операторы» (в большинстве случаев «скрещивание» — crossover и «мутация» — mutation), результатом чего является получение новых решений. Для них также вычисляется значение приспособленности, и затем производится отбор («селекция») лучших решений в следующее поколение.



Этот набор действий повторяется итеративно, так моделируется «эволюционный процесс», продолжающийся несколько жизненных циклов (поколений), пока не будет выполнен критерий остановки алгоритма. Таким критерием может быть:

- нахождение глобального, либо субоптимального решения;
- исчерпание числа поколений, отпущенных на эволюцию;
- исчерпание времени, отпущенного на эволюцию.

Генетические алгоритмы служат, главным образом, для поиска решений в многомерных пространствах поиска.

Таким образом, можно выделить следующие этапы генетического алгоритма:

1. Задать целевую функцию (приспособленности) для особей популяции
2. Создать начальную популяцию
3. (Начало цикла)
4. Размножение (скрещивание)
5. Мутирование
6. Вычислить значение целевой функции для всех особей
7. Формирование нового поколения (селекция)
8. Если выполняются условия остановки, то (конец цикла), иначе (начало цикла).

Создание начальной популяции

Перед первым шагом нужно случайным образом создать начальную популяцию; даже если она окажется совершенно неконкурентоспособной, вероятно, что генетический алгоритм все равно достаточно быстро переведет её в жизнеспособную популяцию. Таким образом, на первом шаге можно особенно не стараться сделать слишком уж приспособленных особей, достаточно, чтобы они соответствовали формату особей популяции, и на них можно было подсчитать функцию приспособленности (Fitness). Итогом первого шага является популяция N , состоящая из N особей.

Размножение (Скрещивание)

Размножение в генетических алгоритмах обычно половое — чтобы произвести потомка, нужны несколько родителей, обычно два.

Размножение в разных алгоритмах определяется по-разному — оно, конечно, зависит от представления данных. Главное требование к размножению — чтобы потомок или потомки имели возможность унаследовать черты обоих родителей, «смешав» их каким-либо способом.

Почему особи для размножения обычно выбираются из всей популяции N , а не из выживших на первом шаге элементов N_0 (хотя последний вариант тоже имеет право на существование)? Дело в том, что главный бич многих генетических алгоритмов — недостаток разнообразия (diversity) в особях. Достаточно быстро выделяется один-единственный генотип, который представляет собой локальный максимум, а затем все элементы популяции проигрывают ему отбор, и вся популяция «забивается» копиями этой особи. Есть разные способы борьбы с таким нежелательным эффектом; один из них — выбор для размножения не самых приспособленных, но вообще всех особей.

Мутации

К мутациям относится все то же самое, что и к размножению: есть некоторая доля мутантов m , являющаяся параметром генетического алгоритма, и на шаге мутаций нужно выбрать mN особей, а затем изменить их в соответствии с заранее определёнными операциями мутации.

Отбор

На этапе отбора нужно из всей популяции выбрать определённую её долю, которая останется «в живых» на этом этапе эволюции. Есть разные способы проводить отбор. Вероятность выживания особи h должна зависеть от значения функции приспособленности $Fitness(h)$. Сама доля выживших s обычно является параметром генетического алгоритма, и её просто задают заранее. По итогам отбора из N особей популяции N должны остаться sN особей, которые войдут в итоговую популяцию N' . Остальные особи погибают.

Задание:

- Создать искомую функцию на участке $[0..10]$, согласно варианту.
- Создать популяцию из 20 особей, со ста признаками, соответствующими отрезку $[0..10]$ (интервал между признаками = 0,1).
- Создать алгоритм, размножения и мутации, способный на каждом шаге удваивать популяцию.
- Уничтожать половину удвоенной популяции, эвклидово расстояние признаков которых максимально удалено от искомой функции.
- Остановить мутационный процесс при достижении эвклидова расстояния = 0,5.
- Вывести признаки особи с наименьшим эвклидовым расстоянием в виде графика и рядом график функции для сравнения.

Варианты искомых функций

№	Функция
1	$Y=\sin(x)$
2	$Y=\cos(x)$
3	$Y=x^2$
4	$Y=\ln(x)$
5	$Y=1+\sin(x)$
6	$Y=\sqrt{x}$
7	$Y=\sin(2x)$
8	$Y=\cos(2x)$
9	$Y=2x^2$
10	$Y=\ln(2x)$
11	$Y=1+\sin(2x)$
12	$Y=\sqrt{2x}$
13	$Y=\sin(3x)$
14	$Y=\cos(3x)$
15	$Y=3x^2$
16	$Y=\ln(3x)$
17	$Y=1+\sin(3x)$
18	$Y=\sqrt{3x}$
19	$Y=\sin(4x)$
20	$Y=\cos(4x)$
21	$Y=4x^2$
22	$Y=\ln(4x)$
23	$Y=1+\sin(4x)$
24	$Y=\sqrt{4x}$