

Методы оптимизации

*Лекция 10. **Эволюционные вычисления.**
Генетические алгоритмы.*

*Селина Елена
Георгиевна*

Эволюционные вычисления

Принятие решений – каждодневная деятельность человека

Принятие решения в большинстве случаев заключается в генерации всех возможных альтернатив решений, их оценке и выборе лучшей среди них. Принять "правильное" решение – значит выбрать такой вариант из числа возможных, в котором с учетом всех разнообразных факторов и противоречивых требований будет оптимизирована некая общая ценность, то есть решение будет в максимальной степени способствовать достижению поставленной цели.

С появлением и развитием вычислительной техники указанные процессы генерации и выбора альтернатив решений стали реализовываться на компьютерах. В связи с этим появились новые задачи формализации и алгоритмизации процесса принятия решений, которые сейчас составляют отдельную большую область исследований.

Эволюционные вычисления

Формализация той или иной поставленной задачи, как правило, предполагает описание всех важных факторов, влияющих на достижение цели, их взаимодействия, ограничительных условий и критерия оценки качества принимаемого решения, на основе которого можно осуществлять выбор между альтернативами. Обычно в качестве критерия оценки выступает некая целевая функция, аргументами которой являются количественные характеристики, описывающие состояние факторов, влияющих на достижение цели в решаемой задаче. При этом решению, приводящему к наилучшему результату, как правило, соответствует экстремальное значение целевой функции, то есть точка ее максимума или минимума.

Таким образом, процесс генерации вариантов решений и выбора наилучшей из полученных альтернатив сводится, в общем случае, к созданию всех возможных комбинаций значений характеристик, влияющих на целевую функцию, и нахождение такой комбинации, которая приводит к ее экстремальному значению.

Эволюционные вычисления

Появление и бурное развитие вычислительной техники лишь подтолкнуло процесс создания и развития новых методов оптимизации, которые позволяли бы решать все новые классы задач и могли быть реализованы на вычислительных машинах.

На сегодняшний день можно выделить три основных типа методов поиска оптимальных решений

- методы, основанные на математических вычислениях,
- перечислительные методы,
- методы, использующие элемент случайности.

Методы, основанные на математических вычислениях, изучены наиболее полно. Они подразделяются на направленные и ненаправленные. Суть ненаправленного метода состоит в том, что локальный экстремум ищется путем решения системы, как правило, нелинейных уравнений. Эта система составляется путем приравнивания градиента целевой функции к нулю (например, метод градиентного спуска или покоординатного спуска). Направленные методы строятся на перемещении от точки к точке в допустимой области, причем направление подобных перемещений связывается с направлением, на которое указывает градиент (например, метод касательных).

Эволюционные вычисления

К недостаткам этих методов можно отнести очень жесткие условия, накладываемые на целевую функцию. Она должна быть дифференцируема на всем пространстве поиска. При формализации современных задач это условие, как правило, соблюсти не удастся. Кроме того, данные методы находят лишь локальные экстремумы целевой функции, тогда как оптимальному решению соответствует только глобальный экстремум. Таким образом, методы поиска оптимальных решений, основанные на математических вычислениях, применимы лишь в случаях гладких, всюду дифференцируемых унимодальных (имеющих один экстремум на пространстве поиска) целевых функций.

Эволюционные вычисления

Перечислительные методы также изучены достаточно подробно и имеют множество видов и форм. Их основная идея состоит в том, что пространство поиска любой задачи можно представить в виде совокупности дискретных точек. Даже если пространство поиска непрерывно, то конечная точность представления чисел в вычислительных машинах позволяет сделать такое допущение. В этом случае поиск решения будет сводиться к перебору всех точек пространства поиска и вычислению в них целевой функции, в одной из которых она, несомненно, примет экстремальное значение. Для реализации непосредственно процесса поиска сейчас разработано большое количество соответствующих алгоритмов.

Недостаток этих методов очевиден. При увеличении размерности пространства поиска (числа аргументов целевой функции) количество точек пространства значительно увеличивается. Это приводит к значительным временным затратам и необходимости применения все более мощной и дорогостоящей вычислительной техники. В то же время размерность решаемых сейчас задач постоянно растет, а время, доступное для принятия решений сокращается. Таким образом, перечислительные методы также применимы для решения все более сужающегося класса задач.

Эволюционные вычисления

Методы, использующие элементы случайности, стали появляться относительно недавно, по мере того как становились очевидными недостатки методов первых двух видов. В основе первого из таких методов лежит случайный поиск в пространстве задачи с сохранением наилучшего полученного результата. Очевидно, что применение такого метода не гарантирует получения оптимального решения. Кроме того, результат работы метода не может быть лучше, чем в случае методов поиска, поскольку в обоих случаях рассматриваются одни и те же дискретные точки пространства поиска задачи.

Вместе с тем следует заметить, что сейчас при решении очень сложных задач основной целью является поиск уже не оптимального, а более "хорошего" решения по сравнению с полученным ранее или заданным в качестве начального. Здесь методы, использующие элемент случайности, получают определенное преимущество перед остальными. Однако даже с такими допущениями непосредственный случайный поиск является малоэффективным. Исследования показали, что внесение в такие методы элементов детерминированности дает значительное улучшение показателей. Одним из типов таких "частично" случайных методов являются эволюционные вычисления.

Эволюционные вычисления

Эволюционные вычисления – термин, обычно используемый для общего описания алгоритмов поиска, оптимизации или обучения, основанных на некоторых формализованных принципах естественного эволюционного процесса. Детерминированность этих методов заключается в моделировании природных процессов отбора, размножения и наследования, происходящих по строго определенным правилам. Основным правилом при этом является закон эволюции: "выживает сильнейший", который обеспечивает улучшение находимого решения. Другим важным фактором эффективности эволюционных вычислений является моделирование размножения и наследования. Рассматриваемые варианты решений могут по определенному правилу порождать новые решения, которые будут наследовать лучшие черты своих "предков".

В качестве случайного элемента в методах эволюционных вычислений может использоваться, например, моделирование процесса мутации. В этом случае характеристики того или иного решения могут быть случайно изменены, что приведет к новому направлению в процессе эволюции решений и может ускорить процесс выработки лучшего решения.

Эволюционные вычисления

История эволюционных вычислений началась с разработки ряда различных независимых моделей эволюционного процесса. Среди этих моделей можно выделить три основные парадигмы:

- Генетические алгоритмы.
- Эволюционные стратегии.
- Эволюционное программирование.

Основное отличие генетических алгоритмов заключается в представлении любой альтернативы решения в виде битовой строки фиксированной длины, манипуляции с которой производятся в отсутствие всякой связи с ее смысловой интерпретацией. То есть в данном случае применяется единое универсальное представление любой задачи.

Эволюционные стратегии, напротив, оперируют объектами, тесно связанными с решаемой задачей. Каждая из альтернатив решения представляется единым массивом численных параметров, за каждым из которых скрывается, по сути, аргумент целевой функции. Воздействие на данные массивы осуществляется, в отличие от генетических алгоритмов, с учетом их смыслового содержания и направлено на улучшение значений входящих в них параметров.

Эволюционные вычисления

В основе направления эволюционного программирования лежит идея представления альтернатив в виде универсальных конечных автоматов, способных реагировать на стимулы, поступающие из окружающей среды. Соответствующим образом разрабатывались и операторы воздействия на них.

Как и всякий метод, использующий элемент случайности, эволюционные вычисления не гарантируют обнаружения глобального экстремума целевой функции (или оптимального решения) за определенное время. Основное их преимущество в том, что они позволяют найти более "хорошие" решения очень трудных задач за меньшее время, чем другие методы. Естественно, эволюционные вычисления не являются оптимальным средством для решения любых задач, поскольку было доказано, что не существует метода поиска, который был бы наилучшим во всех случаях. Тем не менее 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.

В настоящее время наблюдается взаимное проникновение указанных парадигм и их сращивание в единую концепцию эволюционных вычислений. Поэтому далее имеет смысл рассматривать только одну из них.

Генетические алгоритмы

Генетические алгоритмы, являясь одной из парадигм эволюционных вычислений, представляют собой алгоритмы поиска, построенные на принципах, сходных с принципами естественного отбора и генетики. Если говорить обобщенно, они объединяют в себе принцип выживания наиболее перспективных особей – решений и структуризированный обмен информацией, в котором присутствует элемент случайности, который моделирует природные процессы наследования и мутации. Дополнительным свойством этих алгоритмов является невмешательство человека в развивающийся процесс поиска. Человек может влиять на него лишь опосредованно, задавая определенные параметры.

Будучи разновидностью методов поиска с элементами случайности, генетические алгоритмы имеют целью нахождение лучшего, а не оптимального решения задачи. Это связано с тем, что для сложной системы часто требуется найти хоть какое-нибудь удовлетворительное решение, а проблема достижения оптимума отходит на второй план.

Генетические алгоритмы. Основные термины.

Выше было показано, что генетический алгоритм работает с кодами безотносительно их смысловой интерпретации. Поэтому сам код и его структура описываются понятием генотип, а его интерпретация, с точки зрения решаемой задачи, понятием фенотип. Каждый код представляет, по сути, точку пространства поиска. С целью максимально приблизиться к биологическим терминам, экземпляр кода называют хромосомой, особью или индивидуумом. Далее для обозначения строки кода мы будем в основном использовать термин "особь".

Геном – точка пространства поиска, представляющая собой вектор, в котором каждый элемент отвечает за один из аргументов целевой функции.

Особь – экземпляр генома.

Популяция – совокупность особей. Обычно рассматривается фиксированный размер популяции.

Поклоение – новая популяция, полученная после очередного шага генетического алгоритма.

Родители – порождающие особи.

Потомки – порожденные особи. Родительская пара, как правило, порождает пару потомков.

Генетические алгоритмы. Основные термины.

Оператор скрещивания (crossover) – генерация нового генома из геномов двух родителей. Вероятность применения оператора скрещивания является одним из параметров генетического алгоритма.

Оператор мутации – появление новой особи за счет изменения некоторого параметра генома. Вероятность применения оператора мутации является одним из параметров генетического алгоритма.

Оператор отбора – выбор пар родителей для порождения потомков.

Оператор редукции – выбор особей для уничтожения. Отбор и уничтожение определяются на основе «качества» особи, которое определяется значением целевой функции.

Критерий останова алгоритма:

сформировано заданное количество поколений,
популяция достигла заданного «качества»,

достигнут некоторый уровень сходимости, т.е. «качество» особи практически не изменяется или изменяется очень медленно.

Генетические алгоритмы. Этапы работы.

Этап 1. Формирование начальной популяции на основе некоторого случайного закона, с помощью которого выбираются точки в пространстве поиска.

Этап 2. Выбор родителей для размножения с помощью оператора отбора. Вероятность участия особи в отборе определяется по формуле

$$P_i = \frac{f_i}{\sum_{j=1}^N f_j}, \quad i = \overline{1, N},$$

где P_i – вероятность участия i -й особи в отборе, f_i – значение целевой функции для i -й особи, N – размер популяции.

Генетические алгоритмы. Этапы работы.

Этап 3. Создание потомков от выбранных пар родителей с помощью оператора скрещивания (crossover). Обозначим гены из генома размера n как X_i , тогда геном первого родителя будет выглядеть как

$$X_1 X_2 \dots X_k X_{k+1} \dots X_n.$$

Геном второго родителя представим как

$$Y_1 Y_2 \dots Y_k Y_{k+1} \dots Y_n.$$

Перед выполнением оператора скрещивания выбирается случайное значение $k \in [1, n - 1]$ – точка разбиения генома, и соответственно геномы родителей разбиваются на две части. После этого родители обмениваются вторыми частями геномов, т.е. получаем геномы потомков вида

$$\begin{aligned} X_1 X_2 \dots X_k Y_{k+1} \dots Y_n, \\ Y_1 Y_2 \dots Y_k X_{k+1} \dots X_n. \end{aligned}$$

Вероятность применения оператора скрещивания выбирается от 0,9 до 1,0, чтобы обеспечить постоянное появление новых особей. Для перехода в новую популяцию «лучших особей» прежней популяции используют «элитизм». Количество элитных особей определяется по формуле

$$K = (1 - P)N,$$

где N – размер популяции, P – вероятность применения оператора скрещивания.

Генетические алгоритмы. Этапы работы.

Этап 4. Мутация особей нового поколения. Вероятность мутации достаточно мала. Процесс мутации заключается в замене одного элемента генома на другое значение.

Этап 5. Расширение популяции за счет включения потомков в исходное поколение.

Этап 6. Сокращение расширенной популяции до исходного размера с помощью оператора редукции.

Этап 7. Проверка критерия останова работы алгоритма. Если критерий останова не выполнен, то переход к этапу 2. В противном случае – переход к этапу 8.

Этап 8. Поиск лучшей особи в конечной популяции.

Генетические алгоритмы. Поиск максимума одномерной функции

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 функционирования генетического алгоритма

В качестве кода будем использовать двоичное представление аргументов функции. Это положение представляет собой фенотип нашего алгоритма. Сам код будет представлять собой двоичную строку из 5 бит. Это генотип алгоритма. Целевой функцией будет непосредственно сама рассматриваемая функция, аргументом которой является число, чье двоичное представление использует алгоритм.

Определим некоторые характеристики генетического алгоритма. Пусть размер популяции будет 4, вероятность мутации 0,001, сам процесс мутации заключается в инверсии одного из битов строки, выбираемого случайно по равномерному закону. Оператор скрещивания и отбора аналогичны описанным выше. Поскольку задача является простейшей, будем считать, что алгоритм не использует элитизм.

Генетические алгоритмы. Поиск максимума одномерной функции

Пусть на основе равномерного распределения создана исходная популяция из четырех особей, представленная в таблице

№ строки	Код	Значение целевой функции	Вероятность участия в процессе размножения
1	01011	11	11/43
2	10010	18	18/43
3	00010	2	2/43
4	01100	12	12/43

Предположим, что оператор отбора выбрал для производства потомков две пары строк (1, 2) и (2, 4). Работа оператора скрещивания проиллюстрирована в таблице 2. При этом в каждой паре разбиение на подстроки происходит независимо.

Генетические алгоритмы. Поиск максимума одномерной функции

Таблица 2.

№ строки	Родители	Потомки	Значение целевой функции для потомков
1	0 1011	00010	2
2	1 0010	11011	27
2	100 10	10000	16
4	011 00	01110	14

Пусть оператор мутации, несмотря на низкую вероятность, сработал для младшего бита потомка в строке 3, и данный код изменил свое значение с 10000 на 10001.

Таким образом, популяция за счет порожденных потомков расширилась до восьми особей, представленных в таблице 3.

Генетические алгоритмы. Поиск максимума одномерной функции

Таблица 3.

№ строки	Код	Значение целевой функции
Исходная популяция		
1	01011	11
2	10010	18
3	00010	2
4	01100	12
Порожденные потомки		
5	00010	2
6	11011	27
7	10001	17
8	01110	14

Генетические алгоритмы. Поиск максимума одномерной функции

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

Таблица 4.

№ строки	Код	Значение целевой функции	Вероятность участия в процессе размножения
1	10010	18	18/76
2	11011	27	27/76
3	10001	17	17/76
4	01110	14	14/76

Генетические алгоритмы. Поиск максимума одномерной функции

На этом шаг работы генетического алгоритма закончится. Очевидно, что даже за эту 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.

Таким образом, данный пример наглядно иллюстрирует процесс улучшения как популяции в целом, так и наилучшего решения в частности в результате работы генетического алгоритма.

Решение задачи коммивояжера

Задача коммивояжера является классической оптимизационной задачей. Суть ее заключается в следующем. Дано множество из n городов и матрица расстояний между ними или стоимостей переезда (в зависимости от интерпретации). Цель коммивояжера – объехать все эти города по кратчайшему пути или с наименьшими затратами на поездку. Причем в каждом городе он должен побывать один раз и свой путь закончить в том же городе, откуда начал.

Для решения предлагается следующая задача: имеется пять городов, стоимость переезда между которыми представлена следующей матрицей

0	4	6	2	9
4	0	3	2	9
6	3	0	5	9
2	2	5	0	8
9	9	9	8	0

Решение задачи коммивояжера

Для решения задачи применим следующий генетический алгоритм. Решение представим в виде перестановки чисел от 1 до 5, отображающей последовательность посещения городов. А значение целевой функции будет равно стоимости всей поездки, вычисленной в соответствии с вышеприведенной матрицей. Сразу заметим, что одним из оптимальных решений задачи является последовательность 514235 стоимостью 25.

В качестве оператора отбора будем использовать традиционный оператор, применявшийся в предыдущем примере. При этом заметим, что чем меньше значение целевой функции, тем лучше.

Решение задачи коммивояжера

В качестве оператора скрещивания выберем более изощренную процедуру, похожую на двухточечный оператор скрещивания. Поясним его работу на примере. Пусть есть две родительские перестановки (12345) и (34521). Случайно и равновероятно выбираются две точки разрыва. Для примера возьмем ситуацию, когда первая точка разрыва находится между первым и вторым элементами перестановки, а вторая точка – между четвертым и пятым: (1|234|5), (3|452|1). На первом этапе перестановки обмениваются фрагментами, заключенными между точками разрыва: (*|452|*), (*|234|*). На втором этапе вместо звездочек вставляются соответствующие числа из исходной родительской перестановки, начиная со второго числа выделенного фрагмента и пропуская уже имеющиеся в новой перестановке числа. В данном случае в первой перестановке (1|234|5) таким начальным числом является 3, за ним идет 4, которое есть в новой перестановке, и мы его пропускаем, также пропускаем число 5, переходим на начало перестановки и выбираем число 1. В итоге вместо (*|452|*) получаем (34521), аналогично из (3|452|1) и (*|234|*) получаем (52341).

Решение задачи коммивояжера

Оператор мутации будет представлять собой случайную перестановку двух чисел в хромосоме, также выбранных случайно по равномерному закону. Вероятность мутации 0,01.

Размер популяции выберем равным 4.

Исходная популяция представлена в таблице 5.

Таблица 5.

№ строки	Код	Значение целевой функции
1	12345	29
2	21435	29
3	54312	32
4	43125	32

Решение задачи коммивояжера

Пусть для скрещивания были выбраны следующие пары: (1, 3) и (2, 4). В результате были получены потомки, представленные в таблице 6.

Таблица 6.

№ строки	Родители	Потомки	Значение целевой функции для потомков
1	1 23 45	5 43 12	32
3	5 43 12	1 23 54 мутация 13254	28
2	2 143 5	4 312 5	32
4	4 312 5	2 143 5	29

Решение задачи коммивояжера

Пусть для потомка (12354) сработал оператор мутации и обменялись местами числа 2 и 3. В данном случае строка (12354) изменилась и приняла значение (13254). Популяция первого поколения после отсеечения худших особей в результате работы оператора редукции приняла вид, представленный в таблице 7.

Таблица 7.

№ строки	Код	Значение целевой функции	Вероятность участия в процессе размножения
1 (1)	12345	29	28/111
2 (2)	21435	29	28/111
3 (н)	13254	28	29/111
4 (н)	21435	29	28/111

Пусть для получения второго поколения были выбраны следующие пары строк: (1, 4) и (2, 3). И в результате были получены потомки, показанные в таблице 8.

Решение задачи коммивояжера

Таблица 8.

№строки	Родители	Потомки	Значение целевой функции для потомков
1	123 45	214 35	29
4	214 35	123 45	29
2	21 435	13 452	32
3	13 254	21 354	29

Популяция второго поколения после отсеечения худших особей приняла вид, показанный в таблице 9.

Решение задачи коммивояжера

Таблица 9.

№ строки	Код	Значение целевой функции
1 (1)	12345	29
2 (2)	21435	29
3 (3)	13254	28
4 (н)	21354	29

Таким образом, после двух итераций значение целевой функции для лучшего решения изменилось с 29 на 28, среднее значение изменилось с 30,5 до 28,75, а общее качество с 122 до 111. То есть также налицо незначительное, но улучшение популяции.

Решение задачи коммивояжера

Короткое видео с простым объяснением:

https://www.youtube.com/watch?v=ufAHNtZkO_A

Решение задачи коммивояжера

Спасибо за внимание!