

СПОСОБЫ БИНАРНОГО КОДИРОВАНИЯ В ГЕНЕТИЧЕСКОМ АЛГОРИТМЕ

Базанова Е.П.,

научный руководитель канд. техн. наук Панфилов И.А.

Сибирский федеральный университет

В работе рассматривается генетический алгоритм для решения безусловной задачи оптимизации. Исследованы и протестированы несколько способов кодирования решений.

Генетические алгоритмы работают по аналогии с природным эволюционным механизмом. Они оперируют с совокупностью «индивидов», представляющих собой строки, каждая из которых кодирует одно из решений задачи. Популяция решений инициализируется случайным образом. Приспособленность индивида оценивается с помощью специальной функции. Используя оператор селекции, выбираются наиболее приспособленные решения и получают шанс скрещиваться и давать потомство – получать новые решения. Наихудшие индивиды удаляются и не дают потомства. Новые решения (потомки) подвергаются мутации. Таким образом, приспособленность нового поколения в среднем выше предыдущего.

Рассмотрим основные операторы генетического алгоритма.

Оператор селекции. На этом этапе отбирается популяция для дальнейшего размножения.

- Пропорциональная селекция,
- Турнирная селекция,
- Ранговая селекция.

Выбранный тип селекции может применяться с одной из следующих стратегий:

- *Селекция с усечением.* В процессе селекции с усечением с порогом T , только доля T из всех лучших индивидов может быть отобрана, причем в этой доле каждый имеет одинаковую вероятность отбора.
- *Элитарная селекция.* Как минимум одна копия лучшего индивида всегда переходит в следующее поколение.

Оператор скрещивания. Цель этого оператора - распространение хороших генов по популяции.

- Одноточечное скрещивание,
- Двухточечное скрещивание,
- Равномерное скрещивание,
- Равномерное скрещивание по всей (или части) популяции.

Оператор мутации. Изменяет значение гена в хромосоме на любое другое возможное значение. Случайное изменение генов должно осуществляться с низкой вероятностью, обычно в пределах $[0.001; 0.01]$.

Кодирование решений.

Решения в генетическом алгоритме, как правило, представляются в виде бинарной строки. Если исходные данные задачи были представлены вещественными числами, то для работы алгоритма необходимо определить некоторую функцию кодирования и декодирования решений. Существует много способов кодирования решений. Рассмотрим некоторые из них:

- Прямой бинарный код,
- Код Грея,
- Гамма- и дельта-коды Элиаса.

Исследования показали, что для большинства функций генетические алгоритмы будут работать лучше, если закодировать параметры в строку кодом Грея, а не прямым бинарным кодом. Это связано с тем, что соседние числа отличаются в значениях нескольких битов, к примеру, числа 7 и 8 различаются на 4 бита. Прямое бинарное кодирование добавляет дополнительные разрывы, что усложняет поиск. Это можно показать на примере: пусть требуется минимизировать функцию $f(x) = x^2$. Если в популяции изначально преобладали отрицательные хорошие решения, то с большой вероятностью она сойдется к $-1 = 11 \dots 1$. При этом достигнуть глобального минимума будет практически невозможно, поскольку любые изменения одного бита будут приводить к ухудшению решения. При кодировании кодом Грея такой проблемы не возникает.

Младший разряд в последовательности чисел в коде Грея принимает значения 0 и 1, затем следующий старший разряд становится единичным и младший разряд принимает свои значения уже в обратном порядке (0,1).

N	Прямой бинарный код	Код Грея	N	Прямой бинарный код	Код Грея
0	000	000	4	100	110
1	001	001	5	101	111
2	010	011	6	110	101
3	011	010	7	111	100

Гамма- и дельта-коды Элиаса. Данные коды формируются в зависимости от диапазона числа. Допустим, кодируется число n , тогда гамма-кодом Элиаса для числа n будет его обыкновенное бинарное представление, дополненное слева нулями, количество которых на единицу меньше количества битов двоичного представления числа. Дельта-кодами Элиаса называются гамма-коды, в которых унарная часть также закодирована гамма-кодами. То есть для получения дельта-кода нужно закодировать L (количество значащих битов в двоичном представлении числа n) с помощью гамма-кода Элиаса и дописать двоичное представление числа без старшей единицы. Рассмотрим несколько примеров γ и δ кодов Элиаса:

N	γ код	δ код	N	γ код	δ код
0	—	1	8..15	0001xxx	0 0001xxx
1	1	0 1	16..31	00001xxxx	0 00001xxxx
2..3	01x	0 01x	32..63	000001xxxxx	0 000001xxxxx
4..7	001xx	0 001xx			

Каждый символ x соответствует биту в записи $\beta(n - 2^{\lfloor \log n \rfloor})$.

Кодирование данными методами не требует большого количества памяти, а также одинаково быстро выполняется как кодирование, так и декодирование информации.

Можем предположить, что возможен больший охват поискового пространства при использовании Гамма- и дельта-кодов Элиаса, чем при прямом бинарном и Грей кодах, так как в силу своей специфики, они исключают повторение лишних строк уже закодированных значений. На данном этапе моей работы это проверяется эвристическим путем – проводится апробация данных представлений на множестве тестовых задач.