

ВОЗМОЖНОСТИ L-СИСТЕМ ДЛЯ КОДИРОВАНИЯ ЗАДАЧ ПРИ ОПТИМИЗАЦИОННОМ ПОИСКЕ

Слепцов Николай Владимирович,

к.т.н., доцент кафедры ЭиОП Пензенского государственного университета

Мокрушин Виктор Максимович,

к.э.н., доцент кафедры маркетинга Пензенского государственного университета

05.13.17

nbs_nbs@km.ru

Аннотация: Рассмотрены способы эффективного отображения параметров задач в параметры генетического поиска. Рассмотрены возможности L – систем для генерации сложных структур произвольного вида.

Ключевые слова: Генетические алгоритмы, методы поиска, нейронные сети, правила вывода.

L – SYSTEMS FOR SEARCHING METHODS CODING

Sleptsov Nikolai Vladimirovich

Ph.D. of technical Sciences, assistant professor of department EiOP, Penza State University

Mokrushin Viktor Maksimovich

Ph.D. of economic Sciences, assistant professor of department marketing, Penza State University

Abstract: Ways of effectively tasks data representation to genetic algorithms coding are investigated. The possibilities for making complex structures are discussed.

Keywords: Genetic algorithms, searching methods, neural nets, conclusion rules.

Эффективность работы элементов биологических систем вызвала к жизни использование принципов биологической эволюции для оптимизации практически важных систем, нашедших отражение, например, в эволюционных алгоритмах. По сравнению с обычными оптимизационными методами эволюционные алгоритмы имеют следующие особенности: параллельный поиск, случайные мутации и рекомбинации уже найденных удачных решений. Они хорошо подходят как простой эвристический метод оптимизации многомерных, плохо определенных функций [1].

Преимущества эволюционных алгоритмов поставили вопрос о способах формального и целенаправленного управления методами эволюционных и в частности эволюционно – генетических вычислений (ЭГВ). Конкретизация применений данных методов предполагает реализацию 2 составляющих: представление исходной проблемы и обеспечение конструктивности получаемых/генерируемых решений. Обе составляющие тесно связаны друг с другом и поэтому методы решения, применя-

емые в одном случае, востребованы, хотя бы частично, в другом.

Представление исходной проблемы – суть формирование отображения структуры данных проблемы в код ЭГВ [1,2]. В практическом плане здесь речь идет либо о прямом кодировании задачи в генотипе особи либо о кодировании алгоритма получения (точнее - развертывания) решения в фенотипе. Достоинства и недостатки подходов очевидны: в первом случае обеспечивается наглядность и проблемы с манипулированием строками большой размерности, что ограничивает размерность и модульность итоговых решений, во втором случае обеспечивается и размерность, и модульность решений, но платой является отказ от прозрачности представления задачи в генетическом коде и дополнительное преобразование генотипа в фенотип.

Процесс построения любой сложной структуры неизбежно вызывает использование одних и тех же фрагментов, комбинирование которых с различными вариациями и изменением масштабов обеспечивает быстрое конс-

структивное решение проблемы с оптимально эффективным использованием ресурсов. Сложная, фрагментная архитектура, однажды закодированная в процессе эволюции, может неоднократно использоваться для того, чтобы произвести целую систему. Масштабы применения принципа модульности могут быть любыми, соответственно и размеры фрагментов также могут быть любыми.

Наиболее общей формой описания представления системы любой природы, структур, конструкций являются графы – соединения узлов и дуг, поэтому естественно рассматривать в качестве метода описания кодирования методы генерации графов. В связи с тем, что существующие формальные языки описания графов не позволяют в явной форме отражать развитие исследуемых форм (конструкций, растений и т.д.), в качестве возможной альтернативы рассматривались L – системы [3]. L-системы – модель реального мира, основанная на имитации роста и развития биологических объектов. Неформально L-систему можно определить как параллельный строковый переписывающий механизм, являющийся одной из разновидностей грамматик.

По определению, грамматика включает начальную строку и набор правил продукций (вывода). В L-системе все символы в строке при формировании новой строки переписываются параллельно. При присвоении символам строки определенной интерпретации можно обеспечить ее визуализацию, например, кривая Коха, (рис. 1) может быть описана следующей L-системой:

L1: аксиома: F
правило вывода $F \rightarrow F-F++F-F$:

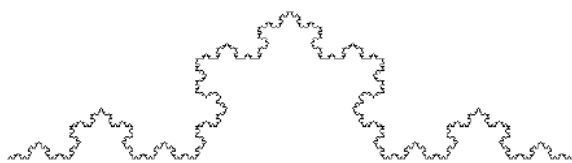


Рис. 1. Кривая Коха

Символы могут интерпретироваться как направление, например, если F будет означать рисование линии, - и + соответственно, левый и правый повороты, то действия при разборе элементов строки будут состоять из рисования линии и поворотов. Величину угла для символов – и +, которая может меняться, задается специальной переменной (δ). Левая часть правила вывода (до \rightarrow) описывает подстроку, которая будет заменена. Могут использоваться любые другие символы, помимо приведенных, так, f может интерпретироваться как перемещение без рисования.

Приведенная интерпретация символов позволяет обеспечить представление только с помощью одиночных линий. Для реальных естественных и технических систем необходимо представление точек ветвления, для которых введем два специальных символа:

$/$ – запомнить текущую позицию и направление перемещения;

\backslash – восстановить последнюю сохраненную позицию и направление.

Приведенные математические конструкции – L-системы при описании формы, роста и развития любых структур принимают генетические, цитологические и физиологические наблюдения во внимание просто как морфологические. Кроме того, L-системы предоставляют возможность описания развитых модульных структур. Они подходят для описания итеративной и дифференциальной модульности. Одно из главных преимуществ L-систем по сравнению с обычными графовыми грамматиками – простота включения контекста, что позволяет задать явную аналогию дифференциации узла (фрагмента, ячейки), например, при росте искусственной нейронной сети (ИНС). Например, для указанной задачи – генерации растущей нейронной сети – применяемый подход основывается на следующей совокупности шагов:

1. ГА генерирует строку битов, хромосому члена популяции. Поиск ГА направлен на получение членов популяции с высокой пригодностью, оцениваемой по данным, получаемым на шаге 3.

2. L-система обеспечивает развитие ИНС, следуя правилам, закодированным в хромосоме. Хромосома декодируется и преобразуется в ряд правил вывода. Они применены к аксиоме для произведения ряда итераций, и получающаяся строка преобразуется в спецификацию структуры для сети.
3. Полученная ИНС обучается для конкретной задачи. Итоговая ошибка преобразуется в оценку пригодности, это значение возвращается ГА.

Контекстно-зависимые L-системы. Для моделирования обмена информацией между соседними объектами применяется контекстное расширение L-систем. Форма правила вывода для контекстно – зависимой L-системы следующая:

$L < P > R \rightarrow S$, где P (предшественник), S (преемник). И L, и R (левый и правый контекст соответственно) могут отсутствовать.

Правило вывода с левым и правым контекстом L и R производит замену P на S, если P предшествует R, либо следует за L. Если к определенному символу одновременно требуется применить два правила вывода, одно – контекстное, другое – бесконтекстное, выбирается контекстное.

Кодирование и переход от сетевого представления к строковому. В соответствии с изложенным, генетический алгоритм должен управлять определенной популяцией, состоящей из наборов правил вывода. Каждый член популяции – двоичная строка, содержащая одно или более правил вывода для L-системы. Для определения пригодности строки производится извлечение из нее правила вывода, далее L-система перезаписывает аксиому, используя эти правила. Получающаяся строка интерпретируется как сеть, которая далее обучается, например, методом обратного распространения, что ограничивает возможную топологию прямонаправленными сетями: все узлы индексируются так, что присутствуют только связи от n_i до n_j для $i < j$.

Способы представления сети можно выбирать различными. Возможным подходом

является представление каждого узла сети произвольным символом алфавита, и использование дополнительных символов ([,], и цифр для определения модульности и соединений. Вариантом такого кодирования является использование неявных связей между соседними символами и применение дополнительных символов для специальных соединений. Для обозначения отсутствия связи между двумя соседними символами применяется запятая (например: A, B). Символы группируются с применением скобок '[]'. Каждая пара скобок связана с определенным уровнем, закодированным числом, при использовании этого числа могут определяться связи с этими группами. Такой способ представления можно назвать абсолютными строками указателей.

Другой подход состоит в применении двоичного алфавита для представления узлов на различных уровнях. Прямые связи между определенными комбинациями 1 и 0 задаются таблицей соединений. Модули задаются размещением узлов в строке на различных уровнях. Узлы в пределах модуля связаны с узлами того же уровня в соответствии с таблицей связи. Такие закодированные строки назовем бинарными табличными строками.

Наконец, возможен способ представления, при котором строки не содержат нумерованных скобок (сами скобки присутствуют), а для указания относительного перехода в пределах строки с целью идентификации определенной связи используются цифры. Варианты представления – соседние символы (они же узлы) связываются по умолчанию, и применяемая запятая используется для обозначения отсутствия такого соединения, либо соседние символы по умолчанию не связаны, и для указания связи между соседними символами применяется знак минус. Такое представление можно назвать строками с относительным пропуском.

Рассмотрим правила вывода (рис. 2). Если в качестве аксиомы берется символ A, процесс вывода может быть описан следующим образом:

- 1: A → BBB
 2: B > B → [C, D]
 3: B → C
 4: C < D → C
 5: D > D → C1

Рис. 2. Типовые правила вывода

Сначала мы имеем аксиому *A* (рис. 3а).

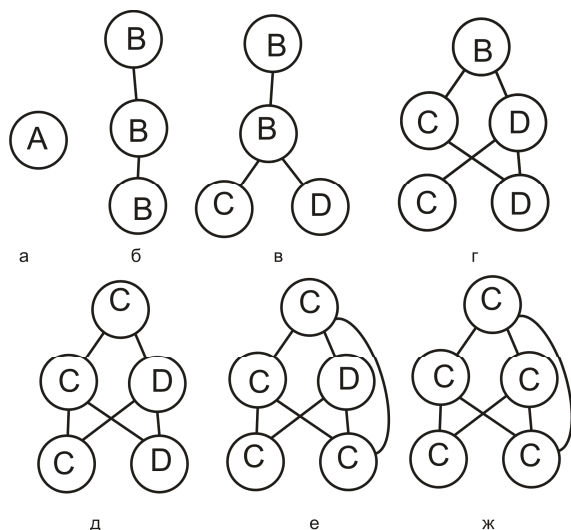


Рис. 3. Преобразования для правил на рис. 2

После первого шага перезаписи получили строку **BBB** (рис. 7б).

В течение второго шага, первый символ **B** перезаписывается, используя правило 2, поскольку первый **B** связан со вторым (рис. 3в). Второй символ **B** также перезаписывается по

правилу 2 (рис. 3г). К третьему символу **B** применяется правило 3, потому что этот символ не связан с другими (есть только связь от другого **B**). В конечном счете это приводит к строке **[C,D][C,D]C**, (рис. 3д). Поскольку вся перезапись проводится параллельно, строки для 3в и 3г не формируются отдельно, а производится непосредственное преобразование сети 3б в 3д. Промежуточные стадии показаны для пояснения.

Третий шаг – первый символ **D** перезаписывается по правилу 5, поскольку первый **D** связан со вторым (рис. 3е), а второй **D** перезаписывается по правилу 4, поскольку первый **C** связан со вторым **D**.

Дальнейшее применение правил вывода невозможно, поэтому конечная строка имеет вид, приведенный на рис. 3ж. Аналогично 3в и 3г, рис. 3е – приведен только для пояснения, в действительности при параллельной перезаписи переход производится из 3д в 3ж.

Описанный подход является лишь одним из механизмов, совместное применение которых обеспечивает высокую эффективность эволюционных вычислений. В частности, в реальных условиях функционирования биологических систем механизм эффективного кодирования всегда дополняется механизмом коррекции, элиминации, многоуровневого отбора и т.д. Тем не менее даже фрагментарное использование эффективных принципов позволяет существенно улучшить показатели работы технических и информационно-логических систем.

Список литературы

1. Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. М.: ФМЛ, 2003.
2. Фогель Л., Оуэнс А., Уолш М. Искусственный интеллект и эволюционное моделирование. М.: Мир, 1969.
3. Kitano H. Designing neural network using genetic algorithm with graph generation system // Complex Systems, 1990, no. 4, pp. 461-476.