УДК 681+004.032.26

ПОИСК АССОЦИАТИВНЫХ ПРАВИЛ В ОПЕРАТОРЕ ГЕНЕРАЦИИ НОВЫХ РЕШЕНИЙ В АЛГОРИТМАХ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

С. В. ЛЮБИМОВ

Северо-Кавказский филиал Белгородского государственного технологического университета им. В.Г. Шухова (г. Минеральные Воды)

e-mail: lysevi@gmail.com

В статье рассмотрены существующие алгоритмы генерации решений в генетическом программировании. Также представлен способ генерации, в котором используются данные, полученные методом поиска ассоциативных правил, что позволяет улучшить сходимость алгоритма. Проведено тестирование алгоритма на примере аппроксимации модельной функции.

Ключевые слова: генетическое программирование, поиск ассоциативных правил.

Идея использования теории эволюции в программировании была предложена еще в работах Фогеля и Рехенберга. Аппарат генетических алгоритмов был впервые введен в работе Холланда. Принципиальное отличие генетических алгоритмов (ГА) от разработанных ранее эволюционных алгоритмов заключалось в использовании оператора скрещивания для получения нового состояния системы[2].

Генетические алгоритмы являются упрощенной моделью эволюции, не претендующей при этом на полноту и достоверность[3]. Решение в генетическом программировании представляется в виде иерархической структуры (рис. 1.), в котором все листья являются термами, а остальные узлы являются функциями.

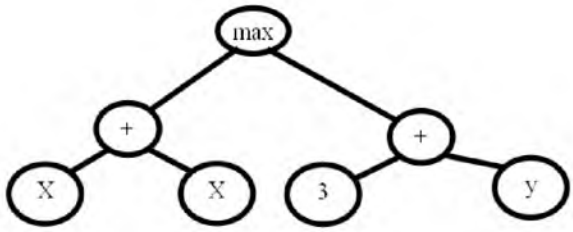


Рис. 1. Дерево арифметического выражения

Существует два метода генерации новых решений: полный и растущий В полном методе узлы выбираются из множества функций пока не достигнута максимальная глубина дерева. В этом алгоритме все узлы на одной и той же глубине, имеют одинаковое число потомков. Все с деревья могут иметь одинаковое количество узлов, если все функции имеют одинаковое количество параметров. Если же функции, используемые при генерации, имеют различное количество параметров, то можно получать программы различного размера и структуры.

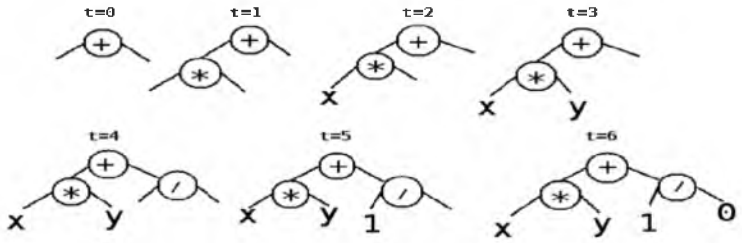


Рис. 2. Пошаговое построение дерева методом full

В растущем алгоритме структура более изменчива. Узлы выбираются из множества, которое содержит как функции, так и терминалы.

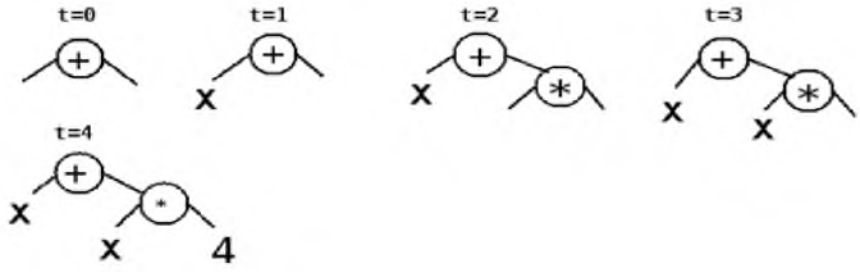


Рис. 3. Пошаговое построение дерева растущим методом

Недостатком обоих алгоритмов в том, что они не учитывают информацию о комбинации вершин, которая положительно сказалась на значении функции оценки.

Цель: разработка алгоритма генерации новых деревьев решений, учитывающего информацию о комбинации вершин, которая положительно сказалась на значении функции оценки.

В последнее время внимание сфокусировано на эволюционных алгоритмах, базирующихся на вероятностной модели. Их называют алгоритмы оценки распределения [4] (Estimation of Distribution Algorithms (EDA)). EDP- популярная модель поиска, в которой распределение данных в популяции оценивается и новые индивиды генерируются по результатам этой оценки.

При анализе предметной области часто возникает последовательность происходящих событий. При обнаружении закономерностей в этих последовательностях можно, с некоторой вероятностью, предсказать их появление в будущем.

Поиск ассоциативных правил есть установление закономерностей между связанными во времени событиями[1]. Метод позволяет на основе анализа поведения временных рядов оценить будущие значения прогнозируемых переменных. Чаще, эти модели должны включать в себя особые свойства времени: иерархию периодов (декада-месяц-год или месяц-квартал-год), положение в цепочке последовательностей, особые отрезки времени (пяти-, шести- или семидневная рабочая неделя, тринадцатый месяц), сезонность, праздники и др.

Анализ рыночных корзин (Basket Analysis) и анализа временных последовательностей являются в настоящий момент одними из самых популярных приложений извлечения данных[1].

Задача поиска ассоциативных правил предполагает нахождение частых наборов в большом числе данных. В контексте анализы рыночной корзины это поиск наборов товаров, которые наиболее часто покупаются вместе. В задаче не учитывался такой атрибут транзакции как время. Тем не менее, взаимосвязь событий во времени также представляет большой интерес.

Поддержкой последовательности S называется отношение количества транзакций, в которое входит последовательность S, к общему количеству транзакций. Последовательность называется частой, если ей поддержка превышает минимальную поддержку, заданную пользователем:

$$Supp(S) > Supp_{min}.$$

Для расчета поддержки узлов необходимо выделить последовательности в множестве узлов. В данном алгоритме они представляю собой вектор следующего вида:

$$\{K, V_p, V_1, V_2, V_3, \ldots, V_n\},$$

где Vp- вершина-предок, V1-n-вершины потомки, K-глубина, на которой располагается вершина Vp.

Рис. 4. Представление поддерева в виде последовательности

Для оценки поддержки каждого узла берутся не все представители популяции, а лишь та часть, фитнесс которой ближе к определенному значению.

Применение алгоритма поиска ассоциативных правил в операторе генерации новых решений

При добавлении нового значения в хромосому, строится множество вероятных значений H. Выбирается такое значение $h \in H, i = \overline{0,|H|}$, у которого значение фитнесса хромосомы из которой взята эта подпоследовательность наибольшее.

Имея информацию о поддержке той или иной последовательности можно генерировать новые решения таким образом, чтобы лучшие решения находившиеся в популяции на момент подсчета поддержки были добавлены в новую популяцию с меньшим числом измененных узлов.

Для реализации описанного механизма предлагается алгоритм, описанный UML диаграммой (рис. 5).

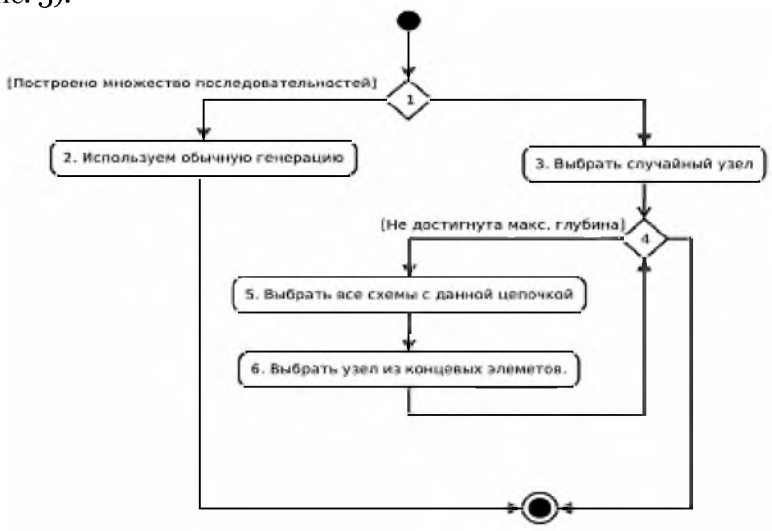


Рис. 5. UML диаграмма алгоритма генерации решений
с использованием информации о поддержке

Алгоритм включает следующую последовательность шагов:

В начале проверяется, было ли построено множество подпоследовательностей ($S \neq \varnothing$). Данное условие может не выполниться, если это генерация первой популяции, и следовательно построение последовательностей еще не проводилось. Если множество найденных последовательностей пусто, то используется один из стандартных генераторов (например «полный» алгоритм).

Используется один из стандартных алгоритмов.

Выбирается случайный узел из множества узлов–функций $f_i \in F, i = \overline{0,|F|}$. Поскольку при построении модели решение не может быть представлено в виде константы или некоторой переменной, то множества используемых констант и переменных не используется при генерации первого узла (корня дерева).

Ограничивает глубину получаемого дерева. Максимальная глубина D_{max}, как и во всех алгоритмах ГП с представлением в виде дерева, ограничивается некоторым максимальным значением.

Строится множество всех допустимых цепочек. В результате получается матрица вида 1. В данной матрице столбцы с 1 по – 1 содержат элементы, которые уже находятся в решении.

Случайно выбирается один из элементов матрицы 1 из столбца n.

$$\begin{vmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & h_{22} & \dots & h_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ h_{m1} & h_{m2} & \dots & h_{mn} \end{vmatrix} \quad (1)$$

Последний столбец матрицы содержит элементы, которые встречаются в популяции в комбинации с элементами, сгенерированными в предыдущие шаги.

В блоке 6 вставляется значение из последнего столбца матрицы $h_{jn}, j = \overline{1,m}$ поддержка строки которой максимальна.

Тестирование предложенного алгоритма

В качестве теста использовалась задача аппроксимации модельной функции. В этом промежутки рассчитывалось значение модельной функции (с шагом 0.25), по полученным данным строилось мат.выражение с помощью тестируемого алгоритма, с помощью которого снова рассчитывалось выходное значение. Ошибка рассчитывалась по формуле:

$$E = \frac{\sum_{i=1}^{N} |y_i - y_i'|}{N}. \quad (2)$$

Использовались следующие модельные функции:

$f(x_1, x_2) = \sin(2x_1/\pi) \cdot \sin(2x_2/\pi)$. Область определения [-5;5]

$f(x_1, x_2) = x_1 \cdot \sin(x_2)$. Область определения [-5;5]

$f(x_1, x_2) = x_1 + x_2 - 5$ Область определения [0;5]

Параметры работы алгоритма: размер популяции - 30; вероятность мутации - 0.9; коэффициент отбора - 0.5. Всего было произведено 25 запусков. Ошибка по каждому запуску приведена на на рисунках 6-8. Результаты тестирования приведены в таблице 1.

Таблица 1

Результаты тестирование разработанного алгоритма

Алгоритм	Средняя ошибка аппроксимации модельной функции		
	1	2	3
Стандартный	0.33	0.22	0.25
Разработанный	0.01	0.022	0.23

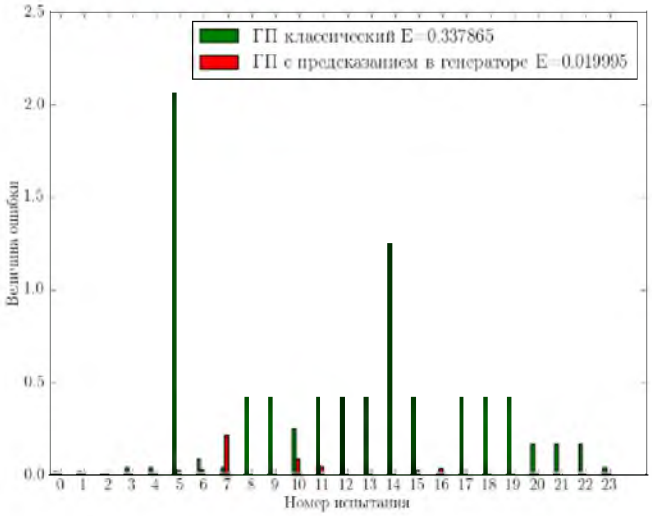


Рис. 6. Результат тестирования разработанного алгоритма на функции $f(x_1, x_2) = \sin(2x_1/\pi) \cdot \sin(2x_2/\pi)$

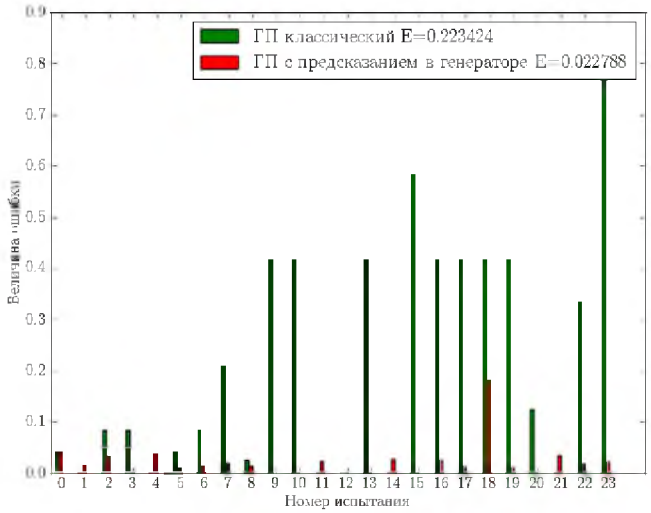
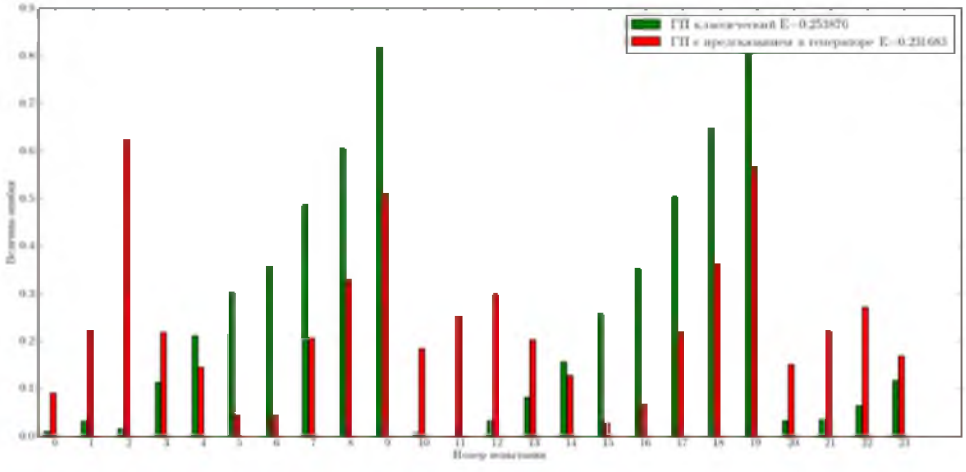


Рис. 7. Результат тестирования разработанного алгоритма на функции $f(x_1, x_2) = x_1 \cdot \sin(x_2)$



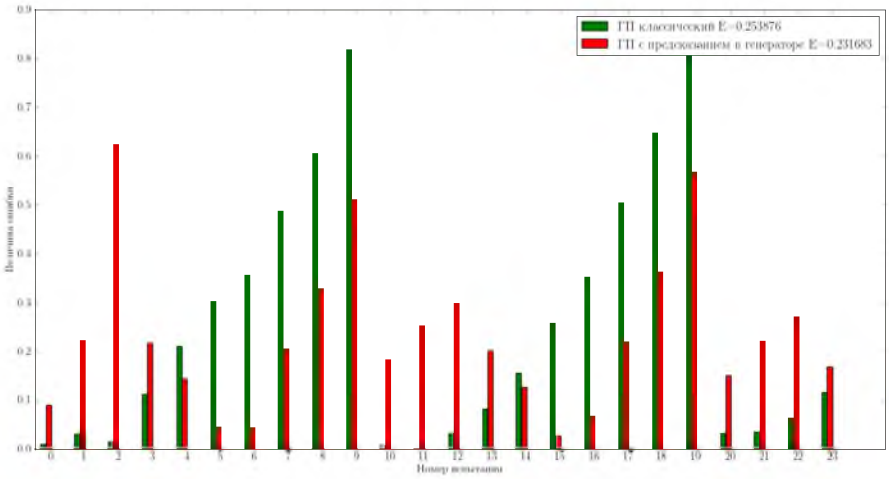


Рис. 8. Результат тестирования разработанного алгоритма на функции $f(x_1, x_2) = x_1 + x_2 - 5$

Выводы

Результаты тестирования показывают, что разработанный метод генерации новых решений для генетического программирования позволяет получить лучшие результаты, чем существующие методы.

Литература

1. Барсегян А.А., Куприянов М.С., Степаненко В.В. Методы и модели анализа данных: OLAP и DataMining[Текст]. Санкт-Петербург: БВХ-Петербург, 2004.

2. Курейчик В.М. Генетические алгоритмы и их применение[Текст]. 2002. С. 245.

3. Шалыто А.А, Царев Ф.Н. Применение генетического программирования для генерации автомата в задаче об «умном муравье» [Текст] // Труды первой Всероссийской научной конференции «Методы и средства обработки информации». URL: http://is.ifmo.ru/genalg/_ant_ga.pdf.

4. Konsuke Y., Hitoshi I. Program Evolution by Integrating EDP and GP[Текст]. URL: http://www.iba.k.u-tokyo.ac.jp/papers/2003/yanaiCECO2004.pdf.

MINING ASSOCIATION RULES IN THE OPERATOR GENERATING NEW SOLUTIONSIN GENETIC PROGRAMMING ALGORITHMS

S. V. LYUBIMOV

North-Caucasian branch Belgorod State Technological University. V.G. Shukhov (City Mineralnye Vody)

e-mail: lysevi@gmail.com

The article deals with the existing algorithms generate solutions in genetic programming. Also shows how the generation that uses the data obtained by mining association rules that can improve the convergence of the algorithm. Tested the algorithm on the example of the approximation of the model function.

Key words: genetic programming, finding association rules.