Интернет-журнал «Науковедение» ISSN 2223-5167 http://naukovedenie.ru/
Том 7, №5 (2015) http://naukovedenie.ru/index.php?p=vol7-5
URL статьи: http://naukovedenie.ru/PDF/136TVN515.pdf
DOI: 10.15862/136TVN515 (http://dx.doi.org/10.15862/136TVN515)

УДК 519.856, 004.8

Кильдюшов Максим Сергеевич
ФГБОУ ВПО «Ухтинский государственный технический университет»
Россия, г. Ухта[1]
Аспирант
E-mail: ItsLastTrue@mail.ru

Программа для восстановления аппроксимированных алгебраических функций от нескольких переменных по набору дискретных значений функции

[1] 169336, Россия, Республика Коми, г. Ухта, ул. Сенюкова 15

Аннотация. В статье представлено описание программного комплекса, являющегося автоматизированной альтернативой математическим методам, используемым для работы с дискретными значениям и сигналами, комплекс создан для поиска сложных многопараметрических, математических зависимостей, в относительно короткие сроки, с помощью синтеза: эвристического алгоритма поиска – генетические алгоритмы, метода безусловной оптимизации функции от нескольких переменных – метод Нелдера-Мида, а также интегрированной Open Source библиотекой MathExpressions.NET упрощающей вид полученных математических выражений. В статье рассматривается возможность использования генетических алгоритмов и генетического программирования восстановления сложных математических зависимостей по заданному набору данных. Рассмотрены конструкции используемых алгоритмов, описаны проблемы их использования (в рамках поставленной задачи), пути решения возникающих проблем, критерии начальной настройки алгоритмов и влияние этих настроек на скорость и эффективность восстановления математических функций различной степени сложности. Предоставлен пример работы программного комплекса описание его структуры, модулей и программной и системной реализации. Автором проведена серьезная работа в сфере разработки автоматизированного программного комплекса, способного восстанавливать аппроксимированные алгебраические функции с определенной точностью по заданному набору дискретных данных.

Ключевые слова: генетические алгоритмы; генетическое программирование; информационные технологии; программирование; стохастическое программирование; методы безусловной оптимизации; системный анализ

Ссылка для цитирования этой статьи:

Кильдюшов М.С. Программа для восстановления аппроксимированных алгебраических функций от нескольких переменных по набору дискретных значений функции // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 7, №5 (2015) http://naukovedenie.ru/PDF/136TVN515.pdf (доступ свободный). Загл. с экрана. Яз. рус., англ. DOI: 10.15862/136TVN515

Введение

В наше время существует множество различных методов, позволяющих работать с дискретными значениями для достижения того или иного результата. Такие методы как:

1. Интерполяция - способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений;
2. Экстраполяция - особый тип аппроксимации, при котором функция аппроксимируется вне заданного интервала, а не между заданными значениями;
3. Аппроксимация - научный метод, состоящий в замене одних объектов другими, в каком-то смысле близкими к исходным, но более простыми.

Когда задача состоит в простом нахождение одного или нескольких значений внутри или за пределами исходных данных методы экстраполяции или интерполяции позволят довольно эффективно решить поставленную задачу, однако в случае если нам необходимо выявить закон изменения y = F (x) при наличии опорных точек задача становится на порядок сложнее. Качество результата напрямую зависит от профессионализма исследователя. Насколько близко он сможет определить приблизительный вид искомого выражения и сможет ли? Не зная даже примерного вида искомой функции, используя стандартные выражения (интерполяционный многочлен Лагранжа, полиномы различной степени, интегральные функции и прочее) остаточный член (разность между заданной функцией и полученной аппроксимирующей) может выдавать слишком высокие значения.

Метод, о котором пойдет речь в данной статье основывается на умном переборе – некий интеллектуальный поиск. В век современных технологий даже самый обычный компьютер обладает настолько высокой производительностью, что даже поиск функций сложного вида занимает не так много времени.

Структура программного комплекса

Программа написана на языке C# для семейства ОС Windows. Метод основывается на эвристическом алгоритме поиска под названием - **генетический алгоритм** (далее - ГА). ГА служит для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе. В данном случае под задачей понимается поиск начального вида математического выражения по набору дискретных значений. Программа создает целые популяции различных математических выражений, проверяет какие из них наиболее близко проходят через заданные условиями задачи точки и, аналогично живой природе, проводит серию скрещиваний и мутаций с самыми лучшими выражениями для получения все новых и новых популяций пока в одной из них не выявится функция, удовлетворяющая поставленной задаче.

Проблема вырождения функций

Основная проблема ГА - недостаток разнообразия, 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. В свою очередь полезные «гены» (в данном случае математические операторы и функции) пропадают, а на первых этапах отбора «малозначительные» гены могут быть полностью заменены. Рассмотрим, например, следующую функцию:

$$y = x^6 + \ln(x),$$

при $x \in [1;5]$ с шагом в единицу.

Таблица 1

Пример влияния операторов функции на её значения

Исходные данные			Значимость			Отношение
x	F(x)		x^6	ln(x)		ln(x)/ x^6
1	1		1	0		0,010830425
2	64,69314718		64	0,69314718		0,001507013
3	730,0986123		729	1,09861229		0,000338451
4	4097,386294		4096	1,38629436		0,000103004
5	15626,60944		15625	1,60943791		0,010830425

Из таблицы 1 видно, что в первую очередь ГА будет пытаться подобрать степенную зависимость и только после того как популяция наполнится множеством подобных математических выражений и конкуренция вырастет алгоритм начнет поиск натурального логарифма. В какой, по счету, популяции начнется подбор малозначимых операторов (с низким влиянием на конечную функцию) событие вероятностное, и наша задача сохранить все математические операторы до этого момента.

После длительных исследований, испытаний и опытов в программе был использован синтез из стандартных методов селекции: рулеточно-островная селекция с заниженным процентом элитарного перехода.

1. Суть **рулеточной** селекции в том, что математические выражения для скрещивания выбираются случайно, но вероятность тем выше чем ближе они к искомой функции относительно других выражений, это усиливает стохастическую составляющую поиска функций;
2. **Островная** селекция заключатся в том, что выражения разделяются на несколько групп скрещивание в которых, происходит только друг с другом, это позволяет увеличить общее разнообразие популяции «давая больше шансов» мало подходящим функциям, но замедляя поиск;
3. **Элитарная** составляющая внесена для того чтобы исключить возможность потери самых близких к искомым функциям выражений в результате скрещивания с мало подходящими и переносить их как эталон из популяции в популяцию – на самых поздних стадиях отбора механизмы скрещивания и мутации все чаще и чаще «портят» функции, особенно если значимость оператора ничтожно мала (как в примере, приведенном в таблице 1).

За счет того, что эти методы противоречат друг другу и дополняют слабые стороны, в рамках поставленной задачи, формируется достаточно быстрый и максимально эффективный комплекс необходимый для аппроксимации простых и сложных математических выражений.

Проблема скорости нахождения решений

С другой стороны, простой перебор, выполняемый даже на очень современных и мощных компьютерах, может занимать недопустимо длительное время, поэтому основной задачей стал поиск методов, позволяющих упростить задачу. К примеру, мы пытаемся восстановить линейную зависимость y = 2*x+7. В популяции мы получили пару похожих выражений (допустим в популяции они числятся под номерами 23 и 78):

$$1)....$$
$$23)\, y = 7 \times x + 5:$$
$$78)\, y = 2 \times x + 7.$$

Программа близка к верному решению и после пары скрещиваний, с некой вероятностью, получит искомое выражение, однако понятно, что искомый вид (линейная функция) уже найден, осталось только подобрать коэффициенты. На помощь приходят методы поиска констант или так называемые методы безусловной оптимизации. Мы имеем многомерное пространство где количество искомых констант соответствует количеству измерений, и идеальный оптимум лежит на пересечении осей. В различных задачах многомерная поверхность исследуемых параметров может иметь множество локальных экстремумов, а оптимум лежать в очень узкой области без существенных градиентов. Наиболее эффективными, из рассмотренных в рамках поставленной задачи, показали себя алгоритм имитации отжига и метод Нелдера-Мида. Алгоритм отжига имеет преимущества по скорости нахождения оптимума при небольшом количестве параметров модели, в то время как метод Нелдера-Мида представляет относительно медленный, но эффективный поиск вплоть до 6 параметров с огромной зависимостью от начальных параметров (коэффициенты растяжения, сжатия, начальный симплекс и другие). С учетом того что в основе мы используем ГА, дающий в конечных популяциях множество похожих функций (а, следовательно, увеличивающий «количество попыток» оптимизации) мы можем позволить себе использовать метод Нелдера-Мида варьируя входящие параметры для поиска наиболее близкого значения. Использование алгоритма на порядок сокращает время поиска функций, а программа позволяет задавать границы, в рамках которых она будет варьировать коэффициенты, пересоздавать начальный симплекс, устанавливать максимальное количество итераций на одну математическую функцию и прочие – то есть программа позволяет регулировать баланс между скоростью поиска и тщательностью отбора.

Проблема упрощения функций

Завершающим этапом работы программы является сокращение функций. В результате работы ГА (а именно процедур скрещивания и мутаций) математические выражения могут довольно быстро разрастаться. К примеру, уже к 30 популяции можно получить математическое выражение следующего вида:

$$F(x) = 0.01 + \mathrm{Cos}(\mathrm{Cos}(\mathrm{Cos}(10.3459 + \mathrm{Cos}(\mathrm{Cos}(\mathrm{Cos}(x) + 10.3459 + \mathrm{Cos}(\mathrm{Cos}(\mathrm{Cos}(x) + x))))))$$
$$+ \mathrm{Pow}(x, 1.39) \times x + 1.39 \times 0.01 + \mathrm{Cos}(x) + \mathrm{Cos}(10.3459 + 1.39 * 0.01)$$
$$+ 10.3459 \times 18.8029 \times \mathrm{Cos}(x)$$

Полученные выражения могут быть достаточно близки к искомым, но в результате перегрузки их вид может кардинально отличаться. Чтобы привести выражения к общему виду в программу была интегрирована Open Source библиотека MathExpressions.NET. В ее основе лежит компиляция выражений в IL код (intermediate language — буквально «промежуточный язык», разработанный фирмой Microsoft для платформы .NET Framework), их преобразование (упрощение, нахождение производных и тому подобное), и дальнейшее возвращение к исходному виду. Чтобы не перегружать поиск вычислительными операциями, автоматическое сокращения происходит только на каждой 5-ой итерации. Этого вполне достаточно чтобы приводить общий вид популяции к упрощенному виду и в случаях поиска сложным математических функций не тормозит процесс.

Восстановление математического выражения с одной переменной

Для тестирования программы возьмем следующую функцию.

$$F(x) = Pow(x,2.39) - 195.50 \times Cos(x)$$

Пусть $x \in [0;25]$, с шагом в единицу, максимально допустимое отклонение равно 1 и через 10 итераций программа выведет промежуточные результаты. Количество особей популяции зададим равным 200 К 10-ой итерации алгоритм подобрал зависимость с отклонением от искомого значения 17.13864 и выделил следующую особь:

$$F(x) = -2 \times Cos(23.4876 + Cos(x)) - Cos(1.39) + Pow(x,2.39) - 193.68470 \times Cos(x)$$

Как видим алгоритм довольно быстро нашел искомую зависимость, но в выражении присутствую лишние гены $2 \times Cos(23.4876 + Cos(x)) + Cos(1.39)$. Установим параметр «максимум итераций» на 100 и продолжим поиск.

В связи с тем, что ГА хоть и является методом «умного» поиска, он все равно имеет стохастический характер, и существует вероятность что избавиться от этих генов вовсе не удастся в этом случае алгоритм Нелдера-Мида, с помощью коэффициентов, сведет влияние лишних генов к минимуму (к примеру, добавит нули или крайне малые значения). Это происходит в следствии того что программа делает упор не на удалении генов, а на поиск их вариаций, что в свою очередь позволяет находить самые сложные математические функции. В дальнейшем, на этапе упрощения функций данные участки математического выражения попадут под сокращения. В результате подобной «взаимовыручки» мы максимально увеличиваем вероятность успеха.

На 27 итерации алгоритм выделил особь с отклонением 0,913961238783563:

$$F(x) = \mathrm{Pow(x,2.39)} - 195.5028 \times \mathrm{Cos(x)}$$

После ручного упрощения:

$$F(x) = -196.5028 \times \mathrm{Cos(x)} + \mathrm{Pow(x,2.39)} + Cos(x)$$

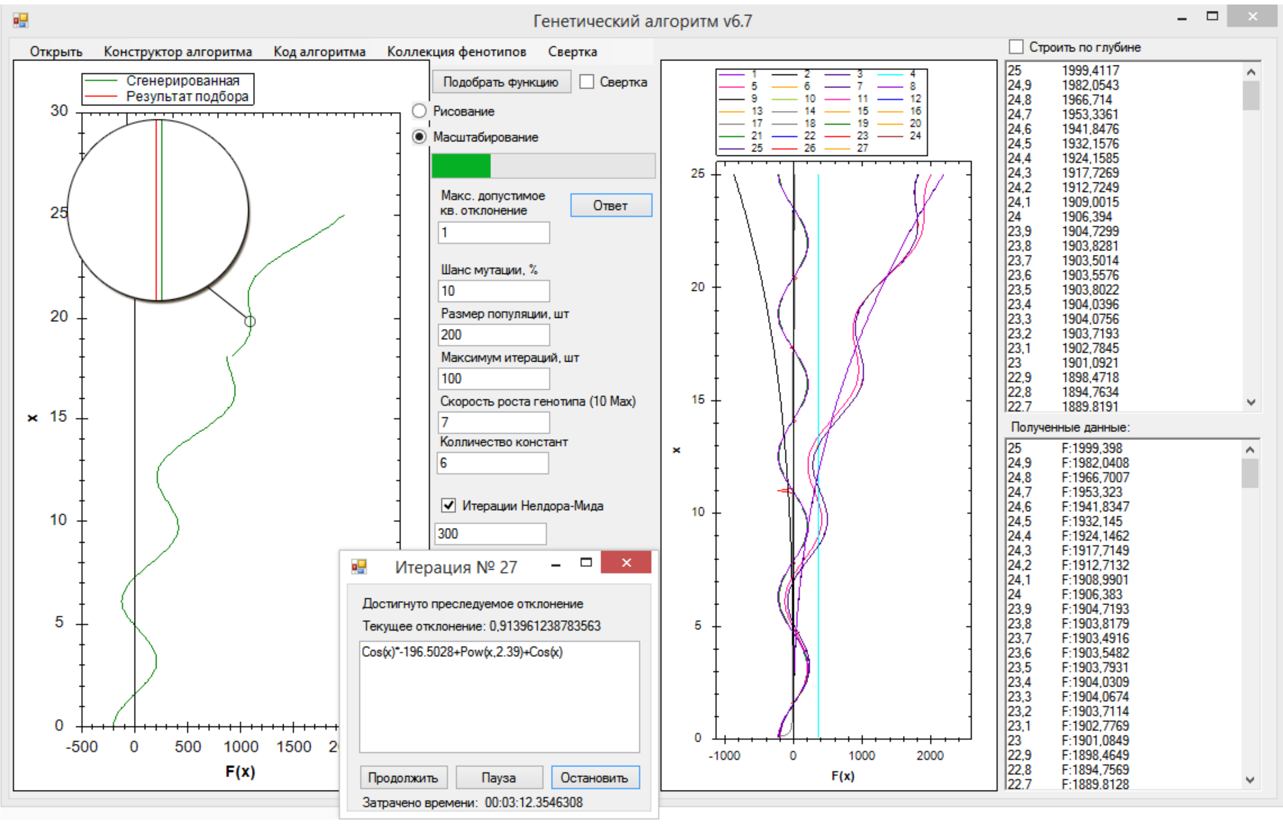

Рисунок 1. Поиск функции с одной переменной

Результат работы программы представлен на рисунке 5, оба графика практически совпадают, отклонение видно только при сильном приближении, итеративный подбор (лучшая особь от каждой популяции) представлен на втором графике этого же рисунка.

Выводы

- ГА являются универсальным методом оптимизации многопараметрических функций, но порождает некоторые недостатки при использовании;
- Комплекс использованных методов сильно зависит от входящих параметров и имеет огромное количество настроек, что позволяет решать широкий спектр задач. Зачастую небольшое изменение одного из них может привести к неожиданному улучшению результата;
- Скорость нахождения искомого выражения сильно зависит от многообразия генов (параметров, функций и констант) используемых в нем, от влияния генов друг на друга и на конечный результат функций. Однако в то же время увеличение количества входящих параметров, увеличение количества операторов и функций (при равномерном влиянии) практически не влияет на скорость подбора;
- Даже с учетом проделанной научной и практической работы и полученных результатов сам комплекс далек от идеала – дальнейшее изучение и апробация методов (и существующих и приведенных в статье) может улучшить программу и увеличить спектр решаемых задач;
- Автоматизация данного процесса (восстановление аппроксимированных функций), в виде альтернативы регрессионным вычислениям, в век развития

информационных технологий, существенно упрощает поставленную задачу и открывает огромные перспективы для дальнейшего развития в различных сферах.

ЛИТЕРАТУРА

1. Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. - М: Физматлит, 2003. - С. 432.
2. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы: Учебное пособие. - 2-е изд. - М: Физматлит, 2006. - С. 320.
3. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы = Sieci neuronowe, algorytmy genetyczne i systemy rozmyte. - 2-е изд. - М: Горячая линия-Телеком, 2008. - С. 452.
4. Веников В.А. Теория подобия и моделирование. М.: Высшая школа, 1968 г.
5. Самарский А.А. Введение в численные методы. М.: Наука. Главная редакция физико-математической литературы, 1982. 272 с.
6. Панченко Т.В. Генетические алгоритмы. Издательский дом «Астраханский университет» 2007. 86 с.
7. Батищев, Д.И. Генетические алгоритмы решения экстремальных задач [Текст] / Д.И. Батищев; Нижегородский госуниверситет. - Нижний Новгород: 1995 – 62 с.
8. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы [Текст] / Под ред. В.М. Курейчика. - 2-е изд., испр. и доп. - М.: ФИЗМАТЛИТ, 2006. - 320 с. - ISBN 5-9221-0510-8.
9. Акимов О.Е. Дискретная математика. Логика, группы, графы. - 2-е изд. - М., Лаборатория базовых знаний, 2001. - 376 с. - "Технический университет".
10. Самофалов К.Г., Романкевич А.М., Валуйский В.Н., Каневский Ю.С., Пиневич М.М. Прикладная теория цифровых автоматов. - К.: Вища школа, 1987. - 375 с.

Рецензент: Статья рецензирована членами редколлегии журнала.

Kildyushov Maksim Sergeevich
Ukhta State Technical University
Russia, Ukhta
E-mail: ItsLastTrue@mail.ru

Abstract. The article describes the software package, which is an automated alternative to the mathematical methods used to work with discrete values and signals, the complex is created to find complex multivariable, mathematical relationships, in a relatively short time, using a synthesis: a heuristic search algorithm - genetic algorithms, methods of unconditional optimization of functions of several variables - Nelder-Mead method, as well as an integrated Open Source library MathExpressions.NET simplifying the form obtained mathematical expressions. Article by considered the possibility of using genetic algorithms and genetic programming of reducing the complex mathematical relationships with some accuracy for a given set of data. We consider the design of the algorithms, describes the use (as part of the problem), solutions to the problems, the criteria for the initial setup of algorithms and the effect of these settings on the speed and effectiveness of recovery of mathematical functions of varying complexity. The design of algorithms and the selection criteria used due to their initial settings. Author of a major operation in the development of automated software system capable of reducing approximated algebraic functions with some accuracy for a given set of discrete data.

Keywords: genetic algorithms; genetic programming; information technology; programming; stochastic programming; unconstrained optimization methods; systems analysis.

REFERENCES

1. Emel'yanov V.V., Kureychik V.V., Kureychik V.M. Teoriya i praktika evolyutsionnogo modelirovaniya. - M: Fizmatlit, 2003. - S. 432.
2. Gladkov L.A., Kureychik V.V., Kureychik V.M. Geneticheskie algoritmy: Uchebnoe posobie. - 2-e izd. - M: Fizmatlit, 2006. - S. 320.
3. Rutkovskaya D., Pilin'skiy M., Rutkovskiy L. Neyronnye seti, geneticheskie algoritmy i nechetkie sistemy = Sieci neuronowe, algorytmy genetyczne i systemy rozmyte. - 2-e izd. - M: Goryachaya liniya-Telekom, 2008. - S. 452.
4. Venikov V.A. Teoriya podobiya i modelirovanie. M.: Vysshaya shkola, 1968 g.
5. Samarskiy A.A. Vvedenie v chislennye metody. M.: Nauka. Glavnaya redaktsiya fiziko-matematicheskoy literatury, 1982. 272 s.
6. Panchenko T.V. Geneticheskie algoritmy. Izdatel'skiy dom «Astrakhanskiy universitet» 2007. 86 s.
7. Batishchev, D.I. Geneticheskie algoritmy resheniya ekstremal'nykh zadach [Tekst] / D.I. Batishchev; Nizhegorodskiy gosuniversitet. - Nizhniy Novgorod: 1995 – 62 s.
8. Gladkov L.A., Kureychik V.V., Kureychik V.M. Geneticheskie algoritmy [Tekst] / Pod red. V.M. Kureychika. - 2-e izd., ispr. i dop. - M.: FIZMATLIT, 2006. - 320 s. - ISBN 5-9221-0510-8.
9. Akimov O.E. Diskretnaya matematika. Logika, gruppy, grafy. - 2-e izd. - M., Laboratoriya bazovykh znaniy, 2001. - 376 s. - "Tekhnicheskiy universitet".
10. Samofalov K.G., Romankevich A.M., Valuyskiy V.N., Kanevskiy Yu.S., Pinevich M.M. Prikladnaya teoriya tsifrovykh avtomatov. - K.: Vishcha shkola, 1987. - 375 s.