зования j-ого режима обслуживания (производства заготовок).

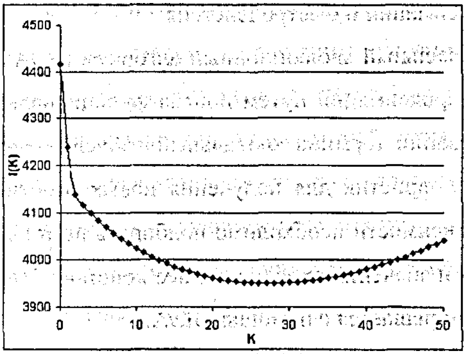
ВМАР-поток задан следующими матрицами интенсивностей переходов (построчно):

$D_0 = ((-3{,}335593;\ 0{,}00038786);(0{,}00038786;\ -0{,}10666141))$;

$D_1 = D_2 = ((1{,}6484036;\ 0{,}01919905);(0{,}0038786;\ 0{,}04925818))$.

Время производства заготовки, время обслуживания требования имеют вырожденное распределение со следующими математическими ожиданиями

$c_1^{(0,1,\dots,K)} = 0{,}1;\ b_1^{(0)} = 0{,}6;\ b_1^{(1,2,\dots,K)} = 0{,}2;\ n^{(0,1,\dots,K)} = 0$.



В результате численного эксперимента было найдено (см. рисунок слева), что функционал качества достигает минимального значения при K=27.

Литература. 1. Lucantoni D. New results on the single server queue with a batch Markovian arrival process. Stochastic Models, 7, 1-46, 1991. **2.** А.Н. Дудин, В.И. Клименок. Системы массового обслуживания с коррелированными потоками. Мн.:БГУ, 2000. **3.** Neuts, M. F. Structured stochastic matrices of M|G|1 type and their applications. Marcel Dekker: New York, USA, 1989.

ЛОКАЛЬНЫЙ ЭВОЛЮЦИОННЫЙ АЛГОРИТМ АППРОКСИМАЦИИ

Киршин Е.А., Юганов А.В., ПГУ, Новополоцк

В настоящее время с развитием вычислительной техники все большее значение приобретает задача аппроксимации, которая представляет собой нахождение функциональной зависимости между входными и выходной переменными на основании известных замеров – наборов значений входных и соответствующих им выходных переменных. С помощью аппроксимации решаются задачи распознавания, прогнозирования, нахождения корней уравнений, решения систем уравнений и др.

В настоящее время существует множество методов решения задачи аппроксимации. Среди них – нейронные сети, генетические (эволюционные) алгоритмы, полиномиальная регрессия, аппроксимация сплайнами, полиномы Чебышева и др. Каждый метод имеет свои преимущества и недостатки. Основным недостатком большинства алгоритмов является их неуниверсальный характер, то есть каждый из них даёт удовлетворительные результаты только при определённых типах аппроксимируемой поверхности или требует модификации в соответствии с ней. Кроме универсальности, к алгоритмам предъявляются также требования точности аппроксимации и быстродействия.

В данной работе представлен локальный эволюционный алгоритм (ЛЭА), обеспечивающий решение задачи аппроксимации путём поиска функциональной зависимости. Присутствие в названии термина «локальный» обусловлено локальным характером применяемых эвристик для получения новых особей. Для нахождения функциональной зависимости необходимо подбирать не только тип функции, т.е. ее структуру, но и значения входящих в нее констант. Эта чрезвычайно сложная задача с успехом решается с помощью ЛЭА.

Для реализации ЛЭА требуется:

1. Определить способ кодирования функциональной зависимости и набор математических операций, используемых в её представлении.

2. Определить набор используемых эвристик.

3. Разработать способ организации особей в популяции.

4. Выбрать способ оценки качества аппроксимации особей-функций.

Как и в традиционных генетических алгоритмах, в данном ЛЭА присутствует популяция, хранящая в себе текущие найденные функции в виде хромосом, объединенных в группы - подпопуляции. Объединение в группы производится по типу функций. Считается, что функции имеют одинаковый тип, если они отличаются только значениями констант. Подпопуляции необходимы для поддержания в популяции определенного числа функций одного типа и предотвращения вхождения поиска констант в локальный оптимум.

В отличие от традиционного генетического алгоритма, в данном ЛЭА делается отступление от представления хромосомы как набора генов, т.е., если рассматривать программную реализацию, в виде некоторой последовательности битов. Функции представляются в виде деревьев, узлы которых – операторы, переменные либо константы. Такое представление функций обеспечивает быструю реализацию алгоритма, т.к. делает более простым осуществление над хромосомами некоторых необходимых эвристик (Grow, Crossover, Mutation – см. далее).

Функции состоят из операций, выбираемых из определенного набора. Этот набор представлен следующими операциями: сложение, вычитание, умножение, деление, экспонента, десятичный логарифм, синус, квадратный корень.

Для оценки подходимости найденных функций создана фитнесс-функция, вычисляющая величину качества аппроксимации Q. Q вычисляется как среднеквадратичное отклонение рассчитанных значений функции в точках замеров от заданных значений функции в точках замеров.

Замеры – это входные данные для алгоритма. Они представляют собой набор значений аргументов и соответствующих им значений функций.

Для ускорения нахождения аппроксимации разработана и внедрена в алгоритм операция сдвига функции (ShiftFunc). Математически можно показать, что среднеквадратичное отклонение минимально, когда среднее отклонение равно нулю. Равенство нулю среднего отклонения можно обеспечить, сдвигая функцию вдоль ее оси на значение среднего отклонения с обратным знаком. Для такого сдвига при каждой функции имеется добавочная константа Const. Т.е. функции имеют следующий вид: $F = f + \mathrm{Const}$, где f – значение, посчитанное по дереву; F – результирующее (выходное) значение.

Одним из важнейших компонентов генетического алгоритма является набор применяемых для получения новых особей эвристик. В данном алгоритме применяются следующие эвристики:

- Мутация узлов и констант (Mutation).

Применяется мутация двух типов: локальная (изменяет только один узел/константу) и глобальная (изменяется сразу несколько констант).

- Доращивание (Grow).

Данная эвристика изменяет произвольно выбранное поддерево

- Скрещивание (Crossover).

Произвольным образом выбираются 2 особи из текущей популяции, затем осуществляется обмен их произвольных поддеревьев, находящихся на одном уровне.

- Рандомная генерация.

Создание функции случайным образом.

Как и генетические алгоритмы в целом, данный эволюционный алгоритм производит итерационное улучшение текущей популяции. На каждой итерации применяются эвристики для получения новых особей и, в зависимости от оценки Q, выданной фитнесс-функцией, производится добавление новой особи в популяцию. В алгоритме имеется множество динамически изменяемых параметров. Например, диапазон изменения констант, критерий добавления особей в популяцию, размер популяции и др.

Исследование ЛЭА производилось в сравнении с нейронными сетями, методом полиномиальной регрессии и визуально – по графику аппроксимирующей поверхности.

В таблице 1 приведены результаты сравнения с нейронной сетью с алгоритмом обратного распространения ошибки, реализованной в программе Delta. В обоих алгоритмах использовались одинаковые методы оценки – среднеквадратичное отклонение.

Таблица 1. Результаты сравнения LGA и Delta

Функция	Q (Delta)	Худшее Q (LGA)	Лучшее Q(LGA)	Среднее Q (LGA)
55.3*X1+2.365*X2+0.58*X3	20526498	4155	$1{,}2*10^{-8}$	2065
X3*X3-(X2+X4)+lg(X4-X1)	13895	0,16	0,05	0,07
X3-X2*(lg(X1))	18966	500	$4{,}01*10^{-26}$	228

В результате исследования обнаружено, что во многих случаях ЛЭА дает лучшее решение в сравнении с другими методами. Таким образом, можно сказать, что ЛЭА является универсальным методом аппроксимации.

Литература. 1. Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. **2.** Ф.Уоссермен, Нейрокомпьютерная техника, М.,Мир, 1992. **3.** S.Grossberg. 1974. Classical and instrumental learning by neural networks. Progress in theoretical biology. **4.** И.А.Минаков. 1999. Сравнительный анализ некоторых методов случайного поиска и оптимизации // Известия Самарского научного центра Российской академии наук, №2, 1999. **5.** Е.Н.Зайцева, Ю.А.Станкевич. Некоторые современные методы решения оптимизационных задач, Материалы Второй международной конференции «Новые информационные технологии в образовании», 1996.

РЕШЕНИЕ ЖЕСТКИХ ЗАДАЧ КОШИ НА ПРИМЕРЕ ЗАДАЧИ РОБЕРТСОНА

Кондратюк А.П., БрГУ, Брест

Модель ROBER, описывающая реакцию Робертсона имеет вид:

$$y_1' = -0.04y_1 + 10^4 y_2 y_3,$$

$$y_2' = 0.04y_1 - 10^4 y_2 y_3 - 3\cdot 10^7 y_2^2,$$

$$y_3' = 3\cdot 10^7 y_2^2,$$

$$y_1(0) = 1,\ y_2(0) = 0,\ y_3(0) = 0,\ x_{out} = 1, 10, 10^2, 10^3, \ldots, 10^{11}.$$

ROBER — один из наиболее известных примеров жестких задач. Эта задача решалась методом Энрайта третьего порядка [1]. Пусть дана задача Коши следующего вида:

$$\begin{cases} y' = f(x,y) \\ y(x_0) = y_0 \end{cases}$$

Рассмотрим эту задачу на отрезке $[x_0; x_{out}]$. Разобьем отрезок точками с одинаковым шагом h. Метод Энрайта третьего порядка имеет следующий вид:

$$y_{n+1} = y_n + h\left(\frac{2}{3}f_{n+1} + \frac{1}{3}f_n\right) - \frac{1}{6}h^2 g_{n+1},$$

где g определяется по правилу дифференцирования сложной функции:

$$y'' = f_x + f_y \cdot f = g(x,y).$$