Оптимизация, идентификация, теория игр

Самонастраивающийся алгоритм генетического программирования для решения задачи Коши и вариационной задачи в символьном виде

Т.С. Карасева[I]

[I]Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет науки и технологий им. академика М.Ф. Решетнева», г. Красноярск, Россия

Аннотация. В статье рассматривается применение самонастраивающегося алгоритма генетического программирования для решения задач символьной регрессии. Обоснована необходимость применения самонастраивающегося типа алгоритма и оператора равномерного скрещивания. Рассмотрено решение задачи Коши для обыкновенных дифференциальных уравнений и вариационной задачи. Данные задачи решаются путем сведения их к процедуре поиска наименьшего значения функции ошибки на множестве символьных выражений. Для такого поиска удобно использовать алгоритм генетического программирования, оперирующий бинарными деревьями, которые кодируют функции-решения, и позволяющий получать точное решение в символьном виде. Для этих задач предложено использовать самонастраивающийся тип алгоритма генетического программирования. Представлены данные численных экспериментов.

Ключевые слова: *алгоритм генетического программирования, самонастраивающийся алгоритм генетического программирования, символьная регрессия, оператор равномерного скрещивания, обыкновенные дифференциальные уравнения, задача Коши, вариационная задача.*

DOI: 10.14357/20790279190307

Введение

При исследовании практически любого процесса предполагается его формализация путем построения математической модели [1]. Но данный способ так же представляет собой сложную задачу, в которой значительная часть усилий направлена на поиск функциональных зависимостей между входными и выходными переменными.

Стоит отметить, что зачастую методы численной идентификации решают задачи путем построения модели, которая, по сути, является черным ящиком [2]. Преодолеть недостаток этих методов позволяет сведение задачи к проблеме символьной регрессии. Задача символьной регрессии состоит в нахождении математического выражения в символьной форме, аппроксимирующего зависимость между начальным и конечным наборами переменных. Алгоритм генетического программирования (ГП) – из класса эволюционных, является эффективным инструментом для решения задач символьной регрессии [3,4].

Однако существует основная проблема, ограничивающая возможности применения ГП. Сложность состоит в необходимости обладания знаниями теории эволюционных алгоритмов при настройке их параметров [5]. Для устранения данной проблемы возможно использовать самонастраивающийся алгоритм генетического программирования (СГП), который самостоятельно выбирает эффективные настройки для решаемой задачи.

1. Алгоритмический комплекс

В ходе исследования был создан алгоритмический комплекс, ядром которого является алгоритм ГП (Рис. 1).

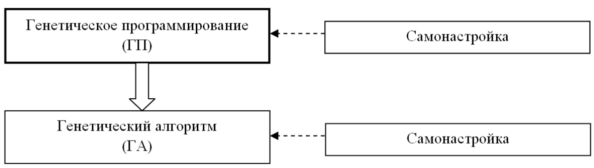
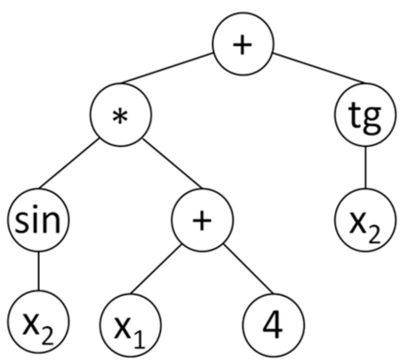


Рис. 1. Алгоритмический комплекс

1.1 Алгоритм генетического программирования

Основной отличительной особенностью алгоритма ГП является представление индивида в виде дерева [6]. Его вершина – это элемент либо функционального, либо терминального множества. Функциональное множество составляют функции, которые используются для формирования решения, а в терминальное множество входит набор констант и всех переменных.

Рассмотрим пример дерева на Рис. 2.



$$F(x_1, x_2) = \sin(x_2) * (x_1 + 4) + tg(x_2)$$

Рис. 2. Пример решения в виде дерева

Представленная в данной работе реализация алгоритма ГП содержит основные группы операторов: селекция (пропорциональная, турнирная и ранговая), скрещивание (стандартное, одноточечное и равномерное), мутация (одноточечная, с заменой ветви).

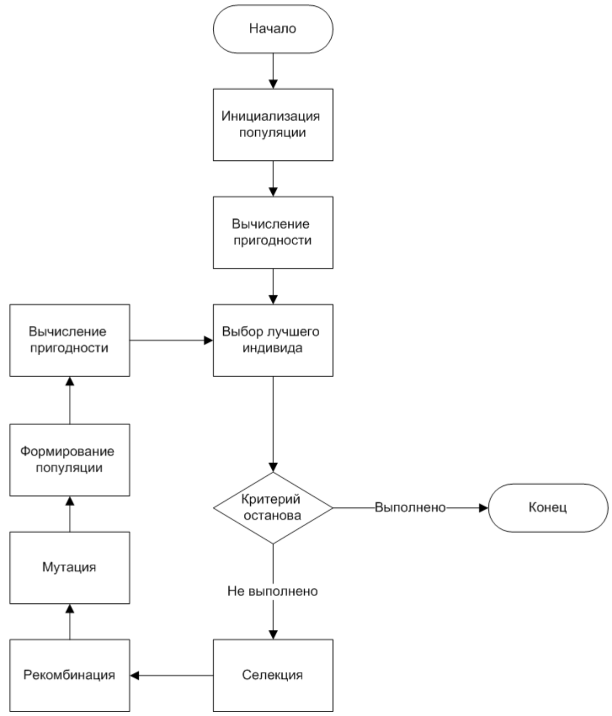


Рис. 3. Этапы алгоритма ГП

Схема последовательности этапов алгоритма на Рис. 3 представляет классический алгоритм ГП.

Для решения задач символьной регрессии при помощи ГП предлагается следующая функция пригодности $fitness$ [3]:

$$fitness = \frac{1}{1 + error},$$

$$error = \frac{1}{n}\sum_{i=1}^{n} \frac{\sqrt{\left(y_i - y_i^*\right)^2}}{max(y^*) - min(y^*)},$$

где n – объем выборки, y_i – значение функции-дерева в i-ой точке, y_i^* – значение целевой функции.

Для оценки эффективности работы алгоритма были выбраны следующие критерии: надежность (доля из общего числа запусков, в которых алгоритм нашел решение) и номер поколения, на котором в среднем было найдено решение.

1.2 Настройка констант зависимости

Функциональная зависимость, в том числе и полученная с помощью алгоритма ГП, всегда содержит некоторые числовые константы. Случайным образом определить нужную константу практически невозможно, поэтому следует использовать методы оптимизации. В разработанной программе присутствует блок оптимизации констант полученного выражения при помощи генетического алгоритма (ГА) [7,8].

1.3 Оператор равномерного скрещивания

Исходя из анализа литературы, был сделан вывод о наиболее частом использовании в алгоритме ГП лишь стандартного и одноточечного операторов скрещивания [6,9]. Однако для ГП возможно реализовать и оператор равномерного скрещивания, обеспечивающий большее разнообразие видов деревьев, который является аналогом одноименного скрещивания в генетических алгоритмах [9]. Сложность его реализации заключается в древовидном представлении индивидов. В результате выполнения равномерного скрещивания создается только один потомок, для которого поддеревья узлов соревнуются между собой, чтобы передать потомку свою вершину. Выбирается определенное число родителей (от двух и более) для последующего скрещивания [10]. Затем у деревьев-родителей, выбираются узлы, один из которых будет передан потомку. Выбор узла осуществляется случайным образом. Стоит отметить, что для каждого узла необходимо проверить его арность.

Если выбранный узел является унарным, то в качестве возможного аргумента одинарной функции могут рассматриваться обе ветви бинарного узла.

Если же осуществляется работа с бинарным узлом, то в качестве возможной варианта левой ветви потомки передается левая ветвь одного из родителей. Аналогично выполняется для правой. Аргумент узла с одинарной функцией передается как возможный вариант в обе ветви бинарного узла.

Рассмотрим пример на Рис. 4. Пусть имеются два дерева-родителя. Пронумеруем узлы слева направо в соответствии с уровнем глубины, на котором он находится.

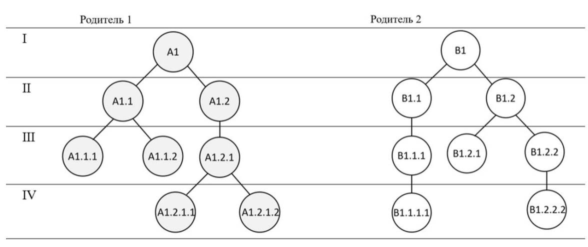


Рис. 4. Родители для реализации равномерного скрещивания

Уточним построение дерева-потомка С. Для этого в Табл. 1 рассмотрим возможные варианты узла для дерева-потомка С.

В результате выполнения равномерного скрещивания для деревьев-родителей из приведенного примера может быть получено следующее дерево С, называемое деревом обобщенных структур (Рис. 5).

Табл. 1

Возможные варианты узла, наследуемые деревом-потомком С

	Узел дерева-потомка	**Возможный вариант узла**
I	С1	А1, В1
II	С1.1	А1.1, В1.1
	С1.2	А1.2, В1.2
III	С1.1.1	А1.1.1, В1.1.1
	С1.1.2	А1.1.2, В1.1.1
	С1.2.1	А1.2.1, В1.2.1, В1.2.2 (если родительский узел унарный)
	С1.2.2	А1.2.1, В1.2.2
IV	С1.1.1.1	В1.1.1.1 (если родительский узел функциональный)
	С1.1.1.2	Нет
	С1.1.2.1	В1.1.1.1 (если родительский узел функциональный)
	С1.1.2.2	Нет
	С1.2.1.1	А1.2.1.1, А1.2.1.2 (если родительский узел унарный), В1.2.2.2
	С1.2.1.2	А1.2.1.2, В1.2.2.2
	С1.2.2.1	А1.2.1.1, А1.2.1.2 (если родительский узел унарный), В1.2.2.2
	С1.2.2.2	А1.2.1.2, В1.2.2.2

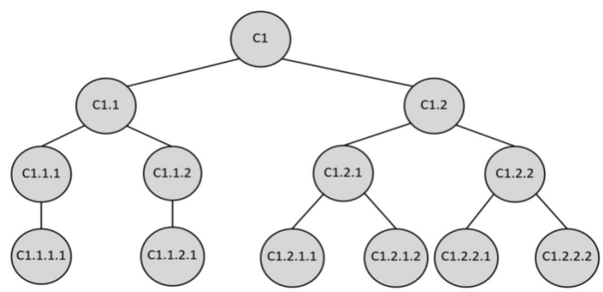


Рис. 5. Дерево обобщенных структур

Для обоснования необходимости включения данного вида скрещивания было проведено тестирование алгоритма ГП с оператором равномерного скрещивания. Был использован набор тестовых функций (Табл. 2), для которых определили настройки из стандартного набора, обеспечивающие лучшую, среднюю и худшую пригодности [11]. Далее такие наборы настроек будем называть «лучшие», «средние» и «худшие» соответственно. В Табл. 3 приведены такие настройки для первой тестовой функции. Для оператора равномерного скрещивания в скобках указано количество родителей, а для оператора турнирной селекции – количество индивидов, участвующих в турнире.

Во всех полученных комбинациях настроек вид скрещивания был заменен на равномерное. Для новых настроек были вновь проведены экспе-

Табл. 2

Тестовые функции для ГП

№	**Тестовая функция**	**Интервал варьирования переменных**
1	$I(x_1, x_2) = x_1^2 + 4x_2^2$	$x_1, x_2 \in [-5; 5]$
2	$I(x_1, x_2) = 0.1x_1^2 + 0.1x_2^2 - 4\,cos(0.8x_1) - 4\,cos(0.8x_2) + 8$	$x_1, x_2 \in [-16; 16]$
3	$I(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$	$x_1, x_2 \in [-2; 2]$
4	$I(x) = 1 - 0.5\,cos(15x - 0.45)\,cos(31.4x) +$ $+0.5\,cos(22.36x)\,cos(35x)$	$x \in [-1; 1]$
5	$I(x) = sin(x)$	$x \in [-\pi; \pi]$

Табл. 3

Результаты тестирования для Задачи 1

№	**Алгоритм ГП**	**Число поколений**	**Надежность**	**Среднее поколение**	**Настройки (селекция, скрещивание, мутация)**
1	Лучшие настройки	300	1	33	Пропорциональная, равномерное (2), сильная
	Средние настройки	300	0.8	44	Турнирная (9), равномерное (7), слабая
	Худшие настройки	300	0.77	49	Пропорциональная, одноточечное, слабая
	СГП	300	0.9	41	Турнирная (2), равномерное (2), средняя

рименты, а полученные результаты (значение надежности, среднее поколение) сравнивались с уже имеющимися. На Рис. 6, 7 представлены усредненные по всем тестовым функциям значения показателей эффективности.

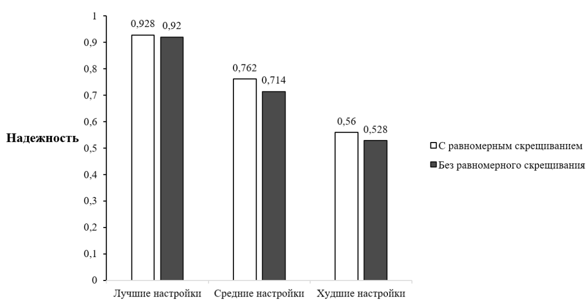


Рис. 6. Надежность алгоритма ГП в зависимости от наличия оператора равномерного скрещивания

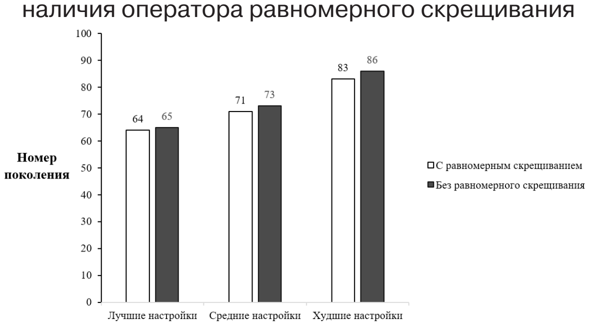


Рис. 7. Номер поколения алгоритма ГП, на котором было найдено решение, в зависимости от наличия оператора равномерного скрещивания

1.4 Метод самонастройки для эволюционных алгоритмов

Для каждой решаемой задачи существует свой, оптимальный в плане эффективности работы алгоритма, набор параметров, который может изменяться во время работы алгоритма. Очевидно, что изначально данные о предпочтительном наборе настроек не известны. Рассматриваемый стандартный алгоритм ГП содержит три группы операторов (Рис. 4)

Табл. 4

Группы операторов базового алгоритма

Группы операторов		
Операторы селекции	Операторы скрещивания	Операторы мутации
Пропорциональная Ранговая Турнирная	Одноточечное Стандартное Равномерное	Одноточечное С заменой ветви

Внутри каждой группы операторы будут конкурировать за возможность быть использованными на следующей итерации алгоритма. Данная самонастройка осуществлялась на уровне популяции [12].

Вероятность выбора того или иного вида оператора из группы зависит от успешности его применения. На первом этапе вероятности выбора вида оператора будут иметь одинаковые значения. На последующих шагах для выбора оператора будет применяться формула:

$$p_i = p_{all} + r_i \frac{(100 - n \cdot p_{all})}{scale},$$

$$error = \sum_{i=1}^{n} \frac{(y_i - y_i^*)^2}{max(y^*) - min(y^*)},$$

где $p_{all} = \frac{20}{n}$, n – число операторов, $r_i = \frac{success_i^2}{used_i}$, $scale = \sum_{j=1}^{n} r_i$, а $used_i$ – число, показывающее,

Табл. 5

Тестовые функции для ГП

№	Тестовая функция	Интервал варьирования переменных
1	$F(x) = 0{,}05(x-1)^2 + \left(3 - 2{,}9e^{-2{,}77x^2}\right)\left(1 - cos\left(x\left(4 - 50e^{-2{,}77x^2}\right)\right)\right)$	$x\epsilon$[-1; 1]$x\epsilon$[-1; 1]
2	$F(x) = 1 - 0{,}5\,cos\big(1{,}5(10x - 0{,}3)\big)\,cos(31{,}4x) +$ $+0{,}5\,cos\big(\sqrt{5} \cdot 10x\big)\,cos(35x)$	$x\epsilon$[-1; 1]$x\epsilon$[-1; 1]
3	$F(x_1, x_2) = 0{,}1x_1^2 + 0{,}1x_2^2 - 4\,cos(0{,}8x_1) - 4\,cos(0{,}8x_2) + 8$	$x_1, x_2\epsilon$[-16; 16]
4	$F(x_1, x_2) = (0{,}1 \cdot 1{,}5x_2)^2 + (0{,}1 \cdot 0{,}8x_1)^2 - 4\,cos(0{,}8 \cdot 1{,}5x_2)^2 -$ $-4\,cos(0{,}8 \cdot 0{,}8x_1) + 8$	$x_1, x_2\epsilon$[-16; 16]
5	$F(x_1, x_2) = 100(x_2 - {x_1}^2) + (1 - x_2)^2$	$x_1, x_2\epsilon$[-2; 2]
6	$F(x_1, x_2) = \dfrac{-10}{0{,}005(x_1^2 + x_2^2) - cos(x_1) \cdot cos\left(\frac{x_2}{\sqrt{2}}\right) + 2} + 10$	$x_1, x_2\epsilon$[-16; 16]
7	$F(x_1, x_2) = \dfrac{-100}{100(x_1^2 - x_2) + (1 - x_1)^2 + 1} + 100$	$x_1, x_2\epsilon$[-5; 5]
8	$\dfrac{1 - sin^2\left(\sqrt{x_1^2 + x_2^2}\right)}{1 + 0{,}001(x_1^2 + x_2^2)}$	$x_1, x_2\epsilon$[-10; 10]
9	$F(x_1, x_2) = 0{,}5(x_1^2 + x_2^2)(2 \cdot 0{,}8 + 0{,}8\,cos(1{,}5x_1)\,cos(3{,}14x_2)$	$x_1, x_2\epsilon$[-5; 5]
10	$F(x_1, x_2) = 0{,}5(x_1^2 + x_2^2)(2 \cdot 0{,}8 + 0{,}8\,cos(1{,}5x_1)\,cos(3{,}14x_2)$	$x_1, x_2\epsilon$[-2,5; $-2{,}5$]
11	$F(x_1, x_2) = x_1^2 \cdot \lvert sin(2x_1)\rvert + x_2^2 \cdot \lvert sin(2x_2)\rvert - \dfrac{1}{5x_1^2 + 5x_2^2 + 0{,}2} + 5$	$x_1, x_2\epsilon$[-4; 4]
12	$F(x_1, x_2) = 0{,}5(x_1^2 + x_1x_2 + x_2^2) \times$ $\times\,(1 + 0{,}5\,cos(1{,}5x_1)\,cos(3{,}2x_1x_2)\,cos(3{,}14x_2) +$ $+0{,}5\,cos(2{,}2x_1)\,cos(4{,}8x_1x_2)\,cos(3{,}5x_2)$	$x_1, x_2\epsilon$[0; 4]
13	$F(x) = sin(x)$	$x\epsilon$[-4,75; 4,75]
14	$F(x_1, x_2) = x_1^2 + 4x_2^2$	$x_1, x_2\epsilon$[-5; 5]
15	$F(x_1, \dots, x_5) = x_1^2 + 4x_2^2 + 9x_3^2 + 16x_4^2 + 25x_5^2$	$x_1, \dots, x_5\epsilon$[-5; 5]
16	$F(x_1, \dots, x_{10}) = x_1^2 + 4x_2^2 + 9x_3^2 + 16x_4^2 + 25x_5^2 + 36x_6^2 +$ $+49x_7^2 + 64x_8^2 + 81x_9^2 + 100x_{10}^2$	$x_1, \dots, x_{10}\epsilon$[-5; 5]
17	$F(x_1, x_2) = (x_1 - 2)^2 + (x_2 - 1)^2$	$x_1, x_2\epsilon$[-5; 5]
18	$F(x) = sin(x)\,x^2 F(x) = sin(x)\,x^2$	$x\epsilon$[-5; 5]
19	$F(x_1, \dots, x_5) = 50 + (x_1^2 - 10\,cos(2\pi \cdot x_1)) + (x_2^2 - 10\,cos(2\pi \cdot x_2)) +$ $+(x_3^2 - 10\,cos(2\pi \cdot x_3)) + (x_4^2 - 10\,cos(2\pi \cdot x_4)) +$ $+(x_5^2 - 10\,cos(2\pi \cdot x_5))$	$x_1, \dots, x_5\epsilon$[-5; 5]

сколько раз был применен i оператор, $success_i$ – количество раз, когда оператор i привел к успеху, то есть пригодность потомка оказалась лучше средней пригодности родительских индивидов.

Описанный подход может использоваться как для ГП, так и для других эволюционных алгоритмов [3,10,12,13].

Целесообразность применения самонастройки доказана результатами тестирования алгоритмов ГП и СГП. Для данного тестирования было использовано репрезентативное множество задач символьной регрессии, тестовые функции которых представлены в Табл. 5 [11].

Стоит отметить, что выбор задач для тестирования алгоритма ГП относится к открытым вопросам [3].

Для поиска решения алгоритмами ГП и СГП было выделено по 100 индивидов, а число поколений равно 300.Число запусков на каждой функции составило 30. Будем говорить, что алгоритм нашел решение, если ошибка составляет менее 1%. Усредненные по 19 задачам результаты экспериментов для ГП и СГП представлены в Табл. 6.

Табл. 6

Результаты оценки эффективности ГП и СГП

Алгоритм		Надежность	Среднее поколение
ГП	Лучшие настройки	0.73	107
	Средние настройки	0.59	120
	Худшие настройки	0.34	138
СГП		0.68	117

Из данных представленной таблицы видно, что значение надежности СГП в среднем лучше, чем соответствующее значение ГП со средними настройками.

Для проверки гипотез о значимости различий использовался критерий Стьюдента. Проверка гипотезы о равенстве математических ожиданий по критерию Стьюдента доказывает статистически значимое превосходство СГП над ГП со средними настройками как для надежности, так и для номера среднего поколения. Эффективность СГП отличается от алгоритма ГП со средними настройками незначительно, т.е. данное различие статистически не значимо.

2. Практическая реализация

Решение обыкновенных дифференциальных уравнений (ОДУ) является неотъемлемой частью многих прикладных задач. Различные научно-технические задачи возможно решить путем сведения их к вариационным. Но методы решения ОДУ и вариационных задач, используемые традиционно, зачастую не позволяют получить решение в символьном виде. Сведение таких задач к процедуре поиска оптимального бинарного дерева, представляющего собой математическую функцию, позволяет использовать алгоритмы ГП [14].

2.1 Решение задачи Коши для обыкновенных дифференциальных уравнений самонастраивающимся алгоритмом генетического программирования.

Существуют два основных метода решения ОДУ: традиционный и численный. Традиционный метод позволяет получить точное, теоретически обоснованное решение в удобной для дальнейшего исследования форме. Однако класс решаемых с помощью традиционных методов уравнений значительно меньше, чем ОДУ решаемых численно. Но результаты численных методов ограничивают возможности дальнейших исследований.

При необходимости поиска решения в символьном виде описанную задачу можно решать путем сведения ее к процедуре поиска наименьшего значения функции ошибки на множестве символьных выражений. При решении данной задачи оптимизации глобальный оптимум равен нулю и достигается на символьном выражении, точно повторяющем истинное решение задачи Коши [15].

В данном исследовании решение задач осуществлялось гибридным методом. На первом этапе осуществляется численное решение ОДУ методом Рунге-Кутты четвертого порядка. В результате решения

Табл. 7

Тестовые задачи Коши для ОДУ

№	Уравнение	Начальные условия	Интервал	Точное решение
1	$xy + (x+1)y' = 0$	$y(0) = 1$	$[0;3]$	$(x+1)e^{-x}$
2	$xy' - 2y - 2x^4 = 0$	$y(-1) = 2$	$[-1;1]$	$x^2 + x^4$
3	$2x(x^2 + y) - y' = 0$	$y(-1{,}4) = 4{,}139327$	$[-1{,}4;1{,}4]$	$e^{x^2} - x^2 - 1$
4	$(y - xy') + x^2 cos(x) = 0$	$y(0{,}1) = 0{,}05998$	$[0{,}1;6]$	$x(0{,}5 + sin(x))$
5	$y'' + 2\,sin(x) = 0$	$y(0) = 0, y'(0) = 2$	$[0;6]$	$2\,sin(x)$

Табл. 8

Решения, полученные алгоритмом ГП, для Задачи 1

Номер запуска	Полученное решение	Номер поколения
1	$((-0{,}999 - \mathrm{x})/(-0{,}999 * \exp(\mathrm{x})))$	25
2	$((0{,}999 + \mathrm{x})/\exp(\mathrm{x}))$	12
3	$((\mathrm{x} + (0{,}21/0{,}21))/\exp(\mathrm{x}))$	38
4	$((1 + \mathrm{x})/(0{,}999 * \exp(\mathrm{x})))$	35
5	$((0{,}999/0{,}999) * (0{,}999/\exp(\mathrm{x})) * (\mathrm{x1} + 0{,}999))$	28
6	$((\mathrm{x} - (-1)) * \exp(((-1) * \mathrm{x})))$	39
7	$(((\exp(\mathrm{x})/\exp(\mathrm{x})) + \mathrm{x})/\exp(\mathrm{x}))$	44
8	$(((\exp(\mathrm{x})/\exp(\mathrm{x})) + \mathrm{x})/\exp(\mathrm{x}))$	34
9	$(((\exp(\mathrm{x})/\exp(\mathrm{x})) + \mathrm{x})/\exp(\mathrm{x}))$	41
10	$((\exp(0{,}992)/\exp((0{,}992 + \mathrm{x}))) * (0{,}992 * (0{,}992 + \mathrm{x})))$	84

ОДУ численным методом будет получена таблица чисел, представляющих значения искомой функции в заданных точках. Данная таблица и является входными данными для алгоритма ГП. На их основании будет найдено выражение в символьном виде.

Задачи, решаемые описанным способом, представлены в Табл. 7 [16]. Для каждой задачи выполнено 10 запусков, по результатам которых определен усредненный номер поколения, на котором было найдено решение, и принадлежность

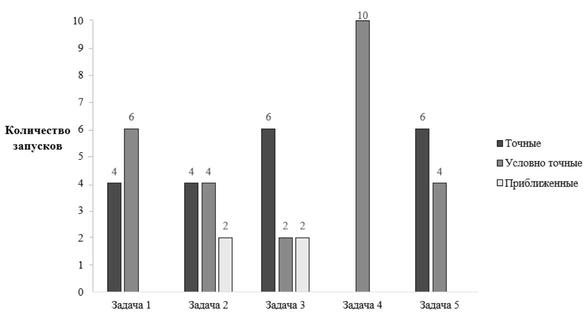


Рис. 8. Символьная точность полученных решений

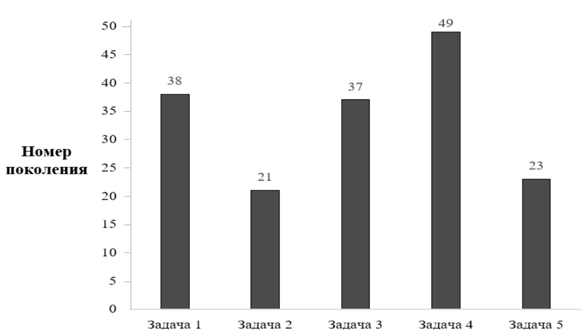


Рис. 9. Усредненный номер поколения, на котором было найдено решение

решения к виду *символьно точные* (решения, приводимые к точным элементарными преобразованиями без округления), *символьно условно точные* (решения, приводимые к точным элементарными преобразованиями с использованием округления) или *приближенные* (решения, требующие более сложных преобразований и/или имеющие сложную громоздкую, не приводимую к точному решению структуру дерева).

Рассмотрим результат работы алгоритмы на примере Задачи 1. В Табл. 8 представлены решения, получаемые алгоритмом, и номер поколения, на котором решение было найдено.

На Рис. 8 представлено количество символьно точных, условно точных и приближенных выражений среди полученных решений для каждой задачи.

Для каждой задачи усредненный номер поколения, на котором было найдено решение, представлены на Рис. 9.

2.2 Решение вариационной задачи самонастраивающимся алгоритмом генетического программирования

Пусть функция $F(x,y,z)$ имеет непрерывные частные производные по всем переменным до второго порядка включительно. Среди всех функций $y(x)$, имеющих непрерывную производную и удовлетворяющую условиям $y(a) = A$, $y(b) = B$ требуется найти такую функцию $y(x)$, которая доставляет экстремум функционалу:

$$I(y(x)) = \int_a^b F(x,y,y')dx.$$

Данную задачу можно решить, используя уравнение Эйлера (обыкновенное дифференци-

Табл. 9

Тестовые вариационные задачи

№	$F(x,y,y')$	Граничные условия	Точное решение
1	$(y')^2+12xy$	X_0=-3, X_N=5; Y_0=-2, Y_N=2	x^3-2x+1
2	$xy'+(y')^2$	X_0=-4, X_N =-1; Y_0=-2, Y_N =4	$-\frac{x^2}{4}+x-1$
3	$(y')^2-2xy$	X_0=1, X_N =-1; Y_0=-3, Y_N =3	$\frac{(7x-x^3)}{6}$
4	$(y')^2x+yy'$	X_0=-2.3, X_N =1.79; Y_0=0.1, Y_N =6	$ln(x)$
5	$(y')^2-y^2$	X_0=-0.7, X_N =-0,28; Y_0=-3, Y_N =3	$1.5\,sin(x)+0.5\,cos(x)$

альное уравнение). Пусть для поставленной задачи получено уравнение Эйлера и граничные условия:

$$F_y-\frac{d}{dx}F_{y'}=0;\ y(x_0)=Y_0,\ y(x_N)=Y_N,$$

где y – искомое решение, y' – производная y, F_y – производная F по y, $F_{y'}$ – производная F по y', x_0 и x_N – границы заданного интервала, Y_0 и Y_N – значения $y(x)$ на границах интервала. Необходимо найти функцию $y(x)$, удовлетворяющую уравнению Эйлера и граничным условиям [17]. Таким образом, данный метод позволяет свести вариационную задачу к краевой (Табл. 9). Для ее решения использовался самонастраивающийся ГП [13]. Поиск решения осуществлялся на основе его оценки по уравнению Эйлера. Индивид в данном случае представляет потенциальное решение вариационной задачи [18]. Модификация ГП состоит в оценке пригодности индивидов. В данной работе функция пригодности индивида представлена следующим образом:

$$Fitness(Ind_k)=\frac{1}{1+E(Ind_k)},$$

$$E(Ind_k)=\frac{\sqrt{\sum_{i=1}^{N}|Ind_k(x_i)|}}{N}+K\left(|y(x_0)-Y_0|+|y(x_N)-Y_N|\right),$$

где $Fitness(Ind_k)$ - значение функции пригодности для k-го индивида; $E(Ind_k)$ – ошибка аппроксимация, вычисляемая по точкам выборки; N – объем выборки; $|Ind_k(x_i)|$ – отклонение от нуля функции F, получаемое при подстановке решения Ind_k в уравнение Эйлера в точках x_i интервала $[x_0;x_N]$; Y_0,Y_N – граничные условия; $y(x_j)$ – значение функции, в точках x_j.

Результат работы алгоритма для Задачи 1 продемонстрирован в Табл. 10.

На Рис. 10 представлено количество символьно точных, условно точных и приближенных реше-

Табл. 10

Решения, полученные алгоритмом ГП, для Задачи 1

Номер запуска	Полученное решение	Номер поколения
1	$x*x*x-2*x+1$	32
2	$x*x*x*\exp(x)/\exp(x)-(x+x)+0{,}999$	74
3	$x*x*x-x-x+\cos(3{,}1)/\cos(3{,}1)$	36
4	$-1{,}999*x+x*x*x*\frac{x}{x}-\frac{2{,}43}{2{,}43}+2$	44
5	$x*x*x-2*x+1/1$	62
6	$4*x*x*x-2{,}999*x*x*x-x*2+0{,}999$	50
7	$(\mathrm{x}*19{,}5+\sin(-4{,}8)-(5{,}4-x*x)+16{,}12*x-\exp(-35)-0{,}2$	58
8	$x*x*x-2*x+1$	28
9	$0{,}999*x*x*x-2*x+1$	47
10	$4*x-x*x*x+2*x*x*x-3*x+0{,}992$	29

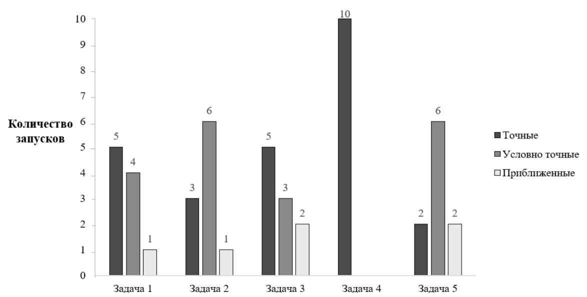


Рис. 10. Символьная точность полученных решений

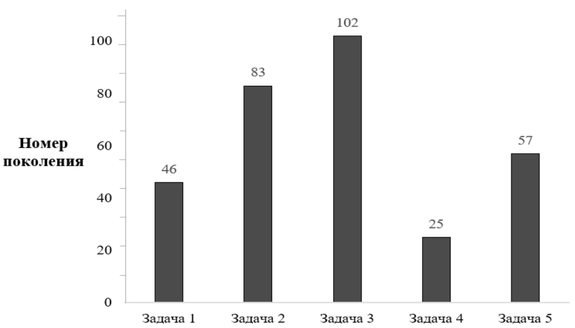


Рис. 11. Усредненный номер поколения, на котором было найдено решение

ний по 10 запускам. Усредненный номер поколения для каждой задачи, на котором было найдено решение, представлен на Рис. 11.

Заключение

В ходе исследования был реализован алгоритмический комплекс, включающий стандартный и самонастраивающийся типы алгоритма генетического программирования для решения задач символьной регрессии, а для настройки коэффициентов выражений применялся генетический алгоритм, для которого так же возможен выбор как стандартного, так и самонастраивающегося типов.

Представлен оператор равномерного скрещивания для алгоритма ГП. Эффективность его применения доказана экспериментально. Ввиду большого числа комбинаций настроек, которое еще возрастает при включении равномерного оператора скрещивания, возникает необходимость автоматизации выбора операторов. Описан возможный вариант самонастройки для эволюционных алгоритмов. Применение данной процедуры позволит значительно сократить временные ресурсы и позволит упростить взаимодействие с алгоритмами исследователей, которые не являются специалистами в области эволюционных алгоритмов.

В рамках данного исследования была осуществлена практическая реализация алгоритмов.

Рассмотрено применение СГП при решении задачи Коши для ОДУ. Данный способ позволяет получать решение в виде точной формулы, если таковая существует, в виде приближенного символьного выражения. Представлено решение вариационной задачи алгоритмом ГП самонастраивающегося типа. Традиционные методы ориентированы на решение определенных типов задач, а ГП позволяет получить решение, когда традиционные методы не могут обеспечить требуемый результат.

Предложенные алгоритмы реализованы в виде программных систем и могут быть использованы в областях науки и техники, где требуется решение непосредственно задачи символьной регрессии, задачи Коши для ОДУ и вариационной задачи.

Литература

1. *Nikolova-Poceva S., Iliev A.* Hybrid Fuzzy Regression Model for Determining Specific Active Power Generation Characteristic of Hydro Power Plants // International Journal on Information Technologies and Security. 2016. Vol. 8, №1. P. 55–68.
2. *Бухтояров В.В., Семенкин Е.С.* Разработка и исследование гибридного метода генетического программирования // Программные продукты и системы. 2010. №3. С. 34-38.
3. *Семенкин Е.С., Семенкина М.Е.* Самоконфигурируемые эволюционные алгоритмы моделирования и оптимизации : монография // МДП, Магнитогорск. 2014. 309 с.
4. *Koza John R.* Genetic Programming: On the Programming of Computers by Means of Natural Selection // MIT Press, Cambridge, MA. 1992. P. 162-169.
5. *Eiben A.E., Smit S.K.* Parameter tuning for configuring and analyzing evolutionary algorithms // Swarm and Evolutionary Computation. 2011. Vol. 1. P. 19-31.
6. *Leung K., Wong M.* Data mining using grammar based genetic programming and applications // New York: Kluwer Academic Publisher. 2002. 213 p.
7. *Goldberg D. E.* Genetic Algorithms in Search, Optimization and Machine Learning // Reading, MA: Addison-Wesley. 1989.
8. *Kazarlis S., Petridis V.* Varying fitness functions in genetic algorithms: studying the rate of increase in the dynamic penalty terms // Proceedings of the 5th International Conference on Parallel Problem Solving from Nature, Berlin, Springer Verlag. 1998. P. 211-220.

9. *Poli R., Langdon W.B.* On the Ability to Search the Space of Programs of Standard, One-Point and Uniform Crossover in Genetic Programming // Technical Report CSRP-98-7. The University of Birmingham (UK). 1998. P. 21
10. *Semenkin E.S., Semenkina M.E.* Self-configuring Genetic Algorithm with Modified Uniform Crossover Operator // Advances in Swarm Intelligence. Lecture Notes in Computer Science 7331. Springer-Verlag, Berlin Heidelberg. 2012. P. 414-421.
11. *Сергиенко А.Б.* Тестовые функции для глобальной оптимизации V.1.32. СибГАУ им. ак. Решетнева. 2015. С. 29-35, 91–95.
12. *Становов В.В., Семенкин Е.С.* Исследование эффективности различных методов самонастройки генетического алгоритма // Актуальные проблемы авиации и космонавтики. 2012. №8. С. 319-320.
13. *Митрофанов С.А., Карасева Т.С.* Решение задач символьной регрессии самонастраивающимся алгоритмом генетического программирования // Актуальные проблемы авиации и космонавтики. 2017. № 2 (13). С. 49-51.
14. *Бураков С.В., Семенкин Е.С.* Решение задачи коши для обыкновенных дифференциальных уравнений методом генетического программирования // Журнал Сибирского федерального университета. Серия: Математика и физика. 2011. Т. 4. № 1. С. 61-69.
15. *Tsoulos I. G.*, *Lagaris I. E.* Solving differential equations with genetic programming // Genet. Program Evolvable. 2006. Vol. 7. P. 33-54.
16. *Филиппов А.Ф.* Сборник задач по дифференциальным уравнениям // НИЦ «Регулярная и хаотическая динамика». Москва–Ижевск. 2003. 235 с.
17. *Гельфанд И.М., Фомин С.В.* Вариационное исчисление, М.: Государственное издательство физико-математической литературы, 1961. 228 с.
18. *Бураков С.В., Семенкин Е.С.* О решении вариационной задачи методом генетического программирования // Сибирский журнал науки и технологий. 2011. №5 (38). С. 19-24.

Карасева Татьяна Сергеевна. ФГБОУ ВО «Сибирский государственный университет науки и технологий им. академика М.Ф. Решетнева», г. Красноярск, Россия. Магистрант. Количество печатных работ: 30. Область научных интересов: эволюционные алгоритмы, генетическое программирование, методы оптимизации. E-mail: tatyanakarasewa@yandex.ru

Self-configuring genetic programming algorithm for a Cauchy problem and variational problem in symbolic form

T.S. Karaseva[I]

[I] Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, Russia

Abstract. The paper considers the application of a self-configuring genetic programming algorithm for solving symbolic regression problems. The necessity to use a self-configuring type of the algorithm and a uniform crossing operator is substantiated. The solution of a Cauchy problem for ordinary differential equations and a variational problem is considered. These problems are solved by reducing them to the procedure for finding the smallest value of the error function on a set of symbolic expressions. For such a search, it is preferable to apply a genetic programming algorithm that operates on binary trees that encode solution functions and allows getting the exact solution in symbolic form. The paper proposes to apply a self-configuring type of a genetic programming algorithm for these problems. The data of numerical experiments are given.

Keywords: *genetic programming algorithm, self-configuring genetic programming algorithm, symbolic regression, operator of the uniform crossover, ordinary differential equations, Cauchy problem, variational problem*.

DOI: 10.14357/20790279190307

References

1. *Nikolova-Poceva S., Iliev A.* Hybrid 2016. Fuzzy Regression Model for Determining Specific Active Power Generation Characteristic of Hydro Power Plants. International Journal on Information Technologies and Security. 8(1): 55–68.
2. *Bukhtoyarov V.V., Semenkin E.S.* 2010. Razrabotka i issledovanie gibridnogo metoda geneticheskogo programmirovaniya [Implementation and investigation of hybryd genetic programming method]. Programmnye produkty i sistemy [Software & systems]. 3:34–38.
3. *Semenkin E.S., Semenkina M.E.* 2014. Samokonfiguriruemye evolyutsionnye algoritmy modelirovaniya i optimizatsii: monografiya [Self-configuring evolution algorithms for modeling and optimization]. MDP, Magnitogorsk. 310 p.
4. *Koza John R.* 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA. P. 162-169.
5. *Eiben A.E., Smit S.K.* 2011. Parameter tuning for configuring and analyzing evolutionary algorithms. Swarm and Evolutionary Computation. 1:19-31.
6. *Leung K., Wong M.* 2002. Data mining using grammar based genetic programming and applications. New York: Kluwer Academic Publisher. 213 p.
7. *Goldberg D. E.* 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley.
8. *Kazarlis S., Petridis V.* 1998. Varying fitness functions in genetic algorithms: studying the rate of increase in the dynamic penalty terms. Proceedings of the 5th International Conference on Parallel Problem Solving from Nature, Berlin, Springer Verlag. 211-220.
9. *Poli R., Langdon W.B.* 1998. On the Ability to Search the Space of Programs of Standard, One-Point and Uniform Crossover in Genetic Programming. Technical Report CSRP-98-7. The University of Birmingham (UK). P. 21
10. *Semenkin E.S. Semenkina M.E.* 2012. Self-configuring Genetic Algorithm with Modified Uniform Crossover Operator. Advances in Swarm Intelligence. Lecture Notes in Computer Science 7331. – Springer-Verlag, Berlin Heidelberg. 414-421.
11. *Sergienko A. B.* 2015. Testovye funktsii dlya global’noy optimizatsii V.1.32 [Test functions for global optimization V.1.32]. SibGAU. 112 p.
12. *Stanovov V.V., Semenkin E.S.* 2012. Issledovanie effektivnosti razlichnykh metodov samonastroyki geneticheskogo algoritma [Investigation of the effectiveness of various methods of self-configuring genetic algorithm]. Aktual’nye problemy aviatsii i kosmonavtiki [Topical Issues in Aeronautics and Astronautics]. 8:319–320.
13. *Mitrofanov S.A., Karaseva T.S.* 2017. Reshenie zadach simvol’noj regressii samonastraivayushchimsya algoritmom geneticheskogo programmirovaniya [Symbolic regression problems solving with self-configuring generic programming algotithm]. Aktual’nye problemy aviatsii i kosmonavtiki [Topical Issues in Aeronautics and Astronautics]. 2 (13):49-51.
14. *Burakov S.V., Semenkin E.S.* 2011. Resheniye zadachi koshi dlya obyknovennykh differentsial’nykh uravneniy metodom geneticheskogo programmirovaniya [Ordinary differential equations Cauchy problem solving with genetic programming techniques] Zhurnal Sibirskogo federal’nogo universiteta. Seriya: Matematika i fizika [Journal of Siberian Federal University. Mathematics & Physics]. 4(1):61-69.
15. *Tsoulos I. G., Lagaris E.* 2006. Solving differential equations with genetic programming. Genet. Program Evolvable. 7:33-54.
16. *Filippov A.F.* 2003. Sbornik zadach po differencial’nym uravneniyam [Collection of problems for differential equations]. Moskva–Izhevsk, NIC «Regulyarnaya i haoticheskaya dinamika» [Regular and chaotic dynamics]. 235 p.
17. *Gelfand I.M., Fomin S.V.* 1961. Variatsionnoye ischisleniye [Calculus of variations] M.: Gosudarstvennoye izdatel’stvo fiziko-matematicheskoy literatury [State Publishing House of Physics and Mathematics]. 228 c.
18. *Burakov S.V., Semenkin E.S.* O reshenii variatsionnoy zadachi metodom geneticheskogo programmirovaniya [On solution of variational problem with genetic programming techniques] Sibirskiy zhurnal nauki i tekhnologiy [Siberian Journal of Science and Technology]. 2011. №5 (38). C. 19-24.

Karaseva T.S. Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, Russia. Graduate student. E-mail: tatyanakarasewa@yandex.ru