

УДК 519.622:004.8.023

С. В. Бураков, Е. С. Семенкин

### О РЕШЕНИИ ВАРИАЦИОННОЙ ЗАДАЧИ МЕТОДОМ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ\*

Для решения вариационной задачи в символьном виде предложено использовать алгоритм генетического программирования. Описаны два варианта поиска решения. Отмечены особенности решения поставленной задачи. Представлены данные численных экспериментов, проведенных с использованием предложенного метода.

*Ключевые слова:* вариационная задача, обыкновенные дифференциальные уравнения, краевая задача, численные методы, алгоритм генетического программирования.

Вариационная задача состоит в нахождении функции, обеспечивающей наибольшее (наименьшее) значение заданного функционала [1]. При решении вариационной задачи обычно пользуются необходимым условием экстремума: с помощью уравнения Эйлера численно решают обыкновенное дифференциальное уравнение (ОДУ), в результате чего обычно получают экстремум функционала (если этот экстремум существует). Для решения ОДУ ставится краевая задача. В общем случае краевая задача не гарантирует существования и единственности решения, но даже если решение существует и единственно, то далеко не каждое решение ОДУ может быть представлено в квадратурах. При численном же решении краевой задачи возникают другие сложности: корректность поставленной задачи, определение устойчивости и сходимости метода и т. д.

В силу того что традиционные методы решения вариационной задачи имеют слабые стороны, необходимо разработать метод решения вариационной задачи, который мог бы объединить преимущества численного и аналитического решения, т. е. позволил бы численно получать функцию в символьном виде, которая являлась бы экстремумом для заданного функционала.

**Постановка задачи.** Пусть задан функционал в следующем виде:

$$J[y(x)] = \int_{x_0}^{x_1} F(x, y(x), y'(x)) dx. \quad (1)$$

Требуется определить функцию  $y(x)$ , доставляющую минимум (максимум) этому функционалу. Предполагается, что на заданном интервале  $[x_0; x_1]$  функционал ограничен, а функция  $y(x)$  дважды непрерывно дифференцируема.

*Примечание 1.* Если функционал задан в виде, отличном от выражения (1), то возможны два варианта решения задачи: преобразовать функционал, т. е. представить его в виде (1), или оставить его в первоначальном виде и изменить метод решения. Для простоты будем предполагать, что функционал может быть приведен к виду (1), хотя нет ограничений и для второго варианта решения задачи.

Для решения вариационной задачи можно воспользоваться эволюционной теорией, а именно алгоритмами генетического программирования (ГП) [2] как инструментом работы с бинарными деревьями.

Данную задачу можно решить двумя методами: первый основан на использовании уравнения Эйлера, т. е. необходимого условия существования экстремума, когда решение ОДУ ищется аналогично [3] (назовем его методом Эйлера), второй ищет функцию, доставляющую наименьшее значение функционала, напрямую (назовем его прямым методом).

**Решение вариационной задачи методом Эйлера.** Этот метод сводит решение вариационной задачи к решению краевой задачи. При этом предполагается, что функция  $F(x, y(x), y'(x))$  из (1) дифференцируема трижды, а искомая функция дифференцируема дважды [1].

Пусть из поставленной задачи получено обыкновенное дифференциальное уравнение (уравнение Эйлера) в виде

$$F_y - \frac{d}{dx} F_{y'} = 0,$$

или

$$F_y - F_{xy'} - F_{yy'} y' - F_{y'y'} y'' = 0, \quad (2)$$

и граничные условия

$$y(x_0) = Y_0, y(x_N) = Y_N, \quad (3)$$

где  $y(x)$  – искомое решение;  $y'$ ,  $y''$  – производные функции  $y(x)$ ;  $Y_0$ ,  $Y_N$  – заданные вещественные константы;  $N$  – объем выборки, т. е. количество точек разбиения отрезка. Требуется найти функцию  $y(x)$ , удовлетворяющую уравнению (2) и граничным условиям (3).

В теории обыкновенных дифференциальных уравнений существование и единственность решения в общем случае не доказаны [4]. Но в поставленной задаче будем считать, что решение существует (для алгоритмов генетического программирования остальные требования: единственности и непрерывности по входным данным – не являются необходимыми для решения задачи).

\*Работа выполнена при поддержке ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2013 годы» (НИР 2011-1.9-519.-005-042) и ФЦП «Научные и научно-педагогические кадры России» (НИР 2011-1.2.1-113-025).

Алгоритм генетического программирования должен оперировать бинарными деревьями (в простейшем варианте), представляющими собой потенциальные решения вариационной задачи. Каждое бинарное дерево состоит из элементов функционального множества (+, −, \*, /, sin, cos, exp, log) и элементов терминального множества ( $x$  или компонентов вектора  $X$ , а также вещественных коэффициентов). В случае решения специфических задач эти множества могут быть дополнены.

В общем виде схему работы алгоритма можно представить следующим образом.

*Шаг 1.* Инициализация начальной популяции (случайным образом).

*Шаг 2.* Оценка пригодности каждого индивида.

*Шаг 3.* Адаптация индивидов.

*Шаг 4.* Применение генетических операторов к имеющимся индивидам для получения новой популяции.

*Шаг 5.* Если критерий остановки не выполнен, то переход к шагу 2.

На шаге 1 начальная популяция инициализируется случайным образом: до заданной начальной глубины случайно выбираются элементы функционального множества и записываются в бинарное дерево, в котором выбираются унарные операторы, и дальнейшее строительство дерева продолжается по одной ветви; на заданной глубине в дерево записываются случайно выбранные элементы терминального множества. Количество индивидов в популяции также является параметром работы алгоритма и должно в некоторой степени зависеть от поставленной задачи, так как слишком большая популяция требует значительных ресурсов для своей обработки, а слишком маленькая не дает точного и надежного результата.

На шаге 2 происходит оценка каждого индивида согласно условиям поставленной задачи, при этом учитываются следующие факторы:

- соответствие функции граничным условиям;
- нарушение тождества при подстановке проверяемого решения задачи в уравнение (2);
- сложность полученной функции.

Соответствие функции граничным условиям определяет значительно более узкий круг функций, подходящих для решения вариационной задачи. При этом возможны три варианта:

- через граничные точки не проходит ни одно решение;
- проходит единственное решение;
- проходит несколько решений.

В первом варианте задача не имеет решения при поставленных краевых условиях. Вторым вариантом является наиболее желанным, когда решение существует и единственно. В третьем варианте нужно найти несколько решений (этот случай будет рассмотрен ниже). Самым простым способом учета граничных условий является вычисление отклонения от них с некоторым коэффициентом. В более сложных случаях может быть введена нелинейная зависимость

функции пригодности от разницы между функцией и точками на границе.

*Примечание 2.* В случае когда решение не единственное и существует конечное число решений, можно последовательно находить решения поставленной задачи, попутно штрафую (например, добавив при вычислении пригодности дополнительное слагаемое) те решения, которые сходятся к уже найденным или известным решениям.

Наличие тождества в уравнении (2) определяет функцию, являющуюся возможным решением задачи (1). Если функционал на этой функции достигает минимума, то уравнение Эйлера превращается в тождество. После проверки всех решений уравнения (2) (если решение не единственное) ищется локальный минимум (максимум) функционала (глобального экстремума функционал может и не достигать).

Нужно учитывать и сложность полученной функции, которая должна быть наименьшей: при идентичности графиков кривых предпочтение отдается функции, которая имеет более короткую, лаконичную запись. Поэтому функцию пригодности можно представить следующим образом:

$$\text{Fit}(P_k) = \frac{1}{1 + E(P_k) + K_1 \cdot D(P_k)}, \quad (4)$$

$$E(P_k) = \frac{\sqrt{\sum_{i=1}^N |P_k(x_i)|}}{N} + K_2 \cdot [ |y(x_0) - Y_0| + |y(x_N) - Y_N| ], \quad (5)$$

где  $\text{Fit}(P_k)$  – значение функции пригодности  $k$ -го индивида;  $E(P_k)$  – ошибка аппроксимации, вычисляемая по всем точкам выборки, здесь  $N$  – объем выборки;  $K_1$  – коэффициент штрафа за сложность дерева;  $D(P_k)$  – число вершин дерева  $P_k$ ;  $K_2$  – коэффициент штрафа за граничные условия;  $|P_k(x_i)|$  – отклонение от нуля (например, среднеквадратическое) функции  $F$  из уравнения (2), получаемое при подстановке решения  $P_k$  в ОДУ в точках  $(x_i)$ , выбранных из интервала равномерно, случайно или каким-либо специальным образом;  $Y_0, Y_N$  – заданные граничные условия,  $y(x_j)$  – значение функции в точке  $x_j$ .

При определении ошибки соответствия решения ОДУ требуется вычисление производных в точках выборки, которое можно выполнить двумя способами: численным (с определенным порядком малости) [5] и аналитическим – путем построения дерева производной функции и нахождения значения этой функции в точках выборки. Так как дифференцирование (в отличие от интегрирования) всегда проводится по строгим правилам, то аналитически вычислить производную не представляет особого труда. При этом нетрудно заметить, что получение численной оценки производной требует меньше машинной памяти и времени, однако аналитически вычисленная производная может быть точнее (при не слишком большом количестве операторов в функции), поэтому ме-

тод вычисления производных должен выбираться с учетом специфики решаемой задачи.

*Примечание 3.* В случае когда через граничные точки проходит не единственное решение, алгоритм может сойтись к какому-то одному решению. Чтобы найти остальные решения, в функцию пригодности, т. е. в знаменатель выражения (4), может быть добавлен дополнительный компонент, например такой:

$$\text{Pen}(P_k) = K_3 \cdot \frac{1}{\sum_{j=1}^M \sqrt{\frac{\sum_{i=1}^N (S_k(x_i) - F_j(x_i))^2}{N}}},$$

где  $K_3$  – коэффициент штрафа за приближение к известному решению;  $M$  – количество частных решений ( $F_j$ ), которые известны (или найдены на предыдущем этапе), и все они не являются искомым решением или являются не всеми искомыми;  $S_k$  – проверяемое решение.

*Примечание 4.* В случае когда функционал представлен в виде, отличном от (1), и не сводится к нему из соображений решения конкретной задачи, функция пригодности может быть представлена в виде, отличном от (4), (5), но обеспечивающем решение задачи. Этот случай требует отдельного исследования, поэтому в данной статье не рассматривается.

На шаге 3 происходит адаптация индивидов, т. е. изменение частей бинарного дерева таким образом, чтобы значение функции пригодности было более высоким. Индивид, лучше адаптирующийся к условиям поставленной задачи, имеет лучшую пригодность, а следовательно, и более высокую вероятность быть отобранным для порождения потомков. С точки зрения теории ОДУ адаптация индивида означает изменение его частей таким образом, чтобы решение имело меньшую погрешность, т. е. означает оптимизацию этого решения – изменение вещественных коэффициентов и/или функционального набора бинарного дерева. Если зафиксировать все вершины дерева кроме какого-то одного узла, то его (дерево) можно рассматривать как функцию одной переменной. Теория оптимизации содержит много методов поиска экстремумов. Применительно к нашей задаче предлагается воспользоваться методом Хука–Дживса, отличающимся простотой и эффективностью [6]. Бинарное дерево (решение ОДУ) в общем случае может содержать много вещественных коэффициентов и узлов-операций, и это делает целесообразным проведение лишь небольшого числа итераций. Для ускорения поиска решения можно варьировать и функциональный набор, содержащийся в дереве, например при случайном изменении функционального термина, сохранить это изменение в случае улучшения или вернуться к старому терму в противном случае.

На шаге 4 применяются генетические операторы селекции, рекомбинации (скрещивания), мутации. Оператор селекции – это оператор отбора индивидов с наибольшей пригодностью. Основная идея селекции состоит в том, что более пригодные индивиды имеют

больше шансов стать родителями нового решения или быть отобранными в новую популяцию. Существуют разные виды селекции: пропорциональная, турнирная, ранговая и др. Скрещивание или клонирование (вероятности скрещивания и клонирования задаются) следуют за селекцией: два отобранных индивида либо скрещиваются, либо один из них клонируется в новую популяцию. При скрещивании два дерева обмениваются поддеревьями, отсеченными в случайно выбранных точках. Это повышает вероятность того, что из деревьев с высокой пригодностью при скрещивании получатся деревья, превосходящие по пригодности родительские. Оператор мутации состоит в случайном изменении одного или нескольких узлов в дереве и рассматривается как средство восстановления генетического разнообразия за счет случайного разброса решения по поисковому пространству.

На шаге 5, если не выполнено условие остановки, следует переход на шаг 2 и повтор цикла. Условием остановки может быть как качественное ограничение – получение решения с заданной точностью, так и количественное – ограниченность ресурсов (главным образом времени).

#### Прямой метод решения вариационной задачи.

При решении задачи (1) прямым методом схема алгоритма генетического программирования остается прежней за исключением действий, связанных с вычислением функции пригодности, поскольку пригодность оценивается по значению функционала.

Если поставлена задача минимизации функционала, то наилучшим решением является функция, которая обеспечивает его минимум. Поэтому функцию пригодности можно представить следующим образом:

$$\text{Fit}(P_k) = \frac{1}{1 + E(P_k) + K_1 \cdot D(P_k) + K_2 \cdot \sum_{j=0}^1 |y(x_j) - Y_j|}, \quad (6)$$

где  $E(P_k)$  – значение функционала на заданном интервале; остальные обозначения соответствуют формулам (4), (5).

В случае если функционал представлен в виде (1), значение  $E(P_k)$  может быть определено численно, например по формуле Симпсона на заданном множестве, когда последовательно находятся значения функции и производных, а затем численное значение интеграла. Полученное таким образом решение доставляет глобальный экстремум функционалу.

Однако если экстремум не достижим, т. е. функционал не ограничен снизу, на заданном множестве, то в этом случае можно говорить лишь о локальном экстремуме, найденном методом Эйлера. При этом апостериорное значение функционала ограничивается только машинной бесконечностью, что и определяет расходимость интеграла и, следовательно, неограниченность функционала.

При прямом решении вариационной задачи в отличие от решения с помощью уравнения Эйлера требуется больше машинных операций в силу того, что

сначала вычисляются значения функции и производной (как и при решении уравнения Эйлера), а потом численно находится значение функционала, что значительно увеличивает количество операций, необходимых для решения задачи, и время работы программы. Однако в отличие от метода Эйлера прямой метод позволяет находить глобальный экстремум, поэтому далее мы будем рассматривать оба метода.

**Результаты численных экспериментов.** Для решения поставленной вариационной задачи была разработана программная система, обеспечивающая возможность решения как прямым методом, т. е. с помощью оценки значения функционала, так и с помощью оценки решения по уравнению Эйлера. При этом общая схема алгоритма отличается только вычислением функции пригодности.

В данной программной системе существует две группы настроек. Первая группа отвечает за текущую поставленную задачу и включает в себя собственно функционал (или ОДУ, представляющее уравнение Эйлера), граничные точки (интервал и граничные условия), функциональный набор, обеспечивающий представление решения. Вторая группа содержит настройки алгоритма генетического программирования: размер популяции и число поколений, ограничивающие максимальное время работы алгоритма, вероятностные коэффициенты скрещивания, клонирования, мутации, вариант отбора (вид селекции). Оба набора настроек влияют на точность и время работы программы, однако на надежность работы алгоритма наибольшее влияние оказывают настройки алгоритма ГП. Настройки задавались таким образом, чтобы соблюдался некий компромисс между точностью, надежностью и временем работы.

Для тестирования алгоритма были решены задачи, взятые из [1; 7] (табл. 1).

Тестовые задачи были решены прямым методом и методом Эйлера. При решении каждым из методов для объективности оценки большинство настроек алгоритма были неизменными, главным образом отличались только вычисленные значения функции пригодности. При этом для каждой задачи осуществлялся 20-кратный прогон алгоритма в силу его стохастичности, а результаты были усреднены. В случае когда функционал оказывался неограниченным (а это 5 из 10 задач), задача решалась только методом Эйлера.

Пример полученного решения приведен в табл. 2. Там же отмечены номер поколения ГП, на котором было впервые найдено это решение, погрешность получаемых решений и время работы алгоритма.

Полученные решения можно разделить на три группы:

– точные решения или решения, приводимые к точным элементарными преобразованиями без округления, например

$$((((x) \cdot (x)) - (2,000)) \cdot (x)) - (-1,000));$$

– условно точные решения, приводимые к точным элементарными преобразованиями с использованием округления, например

$$(((\sin(-4,000) \cdot (x))) + ((\exp(-34,800))/(19,100))) \times (-1,500));$$

– приближенные решения, т. е. решения, требующие более сложных преобразований и/или имеющие сложную громоздкую, не приводимую к точному решению структуру дерева, например

$$61,9/(-39,1) \cdot \sin(\log(31,900) + x).$$

По результатам тестирования были вычислены средние показатели по всем задачам для каждой из групп при решении прямым методом и методом Эйлера (табл. 3).

Таблица 1

Примеры решенных задач

№ п/п	Функция $F(x, y(x), y'(x))$	Граничные условия	Точное решение	Решения, полученные алгоритмом
1	$(y')^2 + 12xy$	-3,0; 5,0 [-2; 2]	$x^3 - 2x + 1$	$(((x \cdot x) - 2,0) \cdot x) - (-1,0)$
2	$(y')^2 - y^2$	-0,7; -0,28 [-3; 3]	$1,5\sin(x) + 0,5\cos(x)$	$61,9/(-39,1) \cdot \sin(\log(31,900) + x)$
3	$(y')^2 + 2yy' - 16y^2$	-1,076; 1,076 [-1,7708 0,585398]	$1,5\sin(4x)$	$((\sin((4,000) \cdot (x))) \cdot (1,500))$
4	$xy' + (y')^2$	-4,0; -1,0 [-2; 4]	$-x^2/4 + x - 1$	$(((x \cdot ((-4) + x)) + 4)/(-4,0))$
5	$(y')^2x^2 + y'$	4,0; -0,667 [0,4; 6]	$2/x - 1$	$(((2,000) - (x))/(x))$
6	$(y')^2x + yy'$	-2,3; 1,79 [0,1; 6]	$\ln(x)$	$(\log(x))$
7	$(y')^2 - 2xy$	1,0; -1,0 [-3; 3]	$(7x - x^3)/6$	$(((x \cdot 45,5) + ((x \cdot x) \cdot (x \cdot (-6,500)))))/39,000)$
8	$(y')^2 - y^2 + 4y \cos(x)$	0,141; 0,7 [-3; 3]	$(x + 2)\sin(x)$	$\log(\exp(\sin(x) \cdot (2,0 + x)))$
9	$(y')^2 - 2xy$	-0,81 2,43 [-2,5; 3,5]	$13x - x^3/6 + 2$	$(((x/(-6,0)) \cdot x \cdot x) + ((x + (2,0)) - (x/(-6,0)))) + x)$
10	$(y')^2 + y^2 + 2ye^x$	-4,19; -0,43[-1; 5]	$1,85914xe^x - 0,5 + 0,5e^{-x}$	$((-4,1) - (((x/((\exp(-17,9)) \cdot (-24,4) + \cos(3,9)))) - \exp(\sin(17,4))) \cdot \exp(0,3))/\exp(x)) + \log(36,6)$

Таблица 2

Результат решения задачи с 20-кратным прогоном

Полученное решение	Номер поколения ГП	Погрешность	Время, мин
$(((((x) \cdot (x)) - (2,000)) \cdot (x)) - (-1,000))$	107	0,09	52
$((1,000) + (((x) \cdot ((x) \cdot (x))) - ((x) + (x))))$	59	0,11	43
$((1,000) - (((x) + (x)) - ((x) \cdot ((x) \cdot (x))))$	48	0,11	21
$(((((1,000) + (((x) \cdot (x)) \cdot (x))) - (x)) - (x))$	103	0,11	53
$(((((x) \cdot (x)) \cdot (x)) + ((1,000) - ((x) + (x))))$	78	0,11	44
$((1,000) - (x)) + (((x) \cdot ((x) \cdot (x))) - (x))$	37	0,11	17
$(((((x) \cdot (x)) - (2,000)) \cdot (x)) + (1,000))$	94	0,09	80
$((x) \cdot (-2,0)) + (((x) \cdot (x)) \cdot (x)) + (\cos((0,0) \cdot (x))))$	27	0,14	20
$((1,000) + (((x) \cdot (x)) \cdot (x))) - ((x) + (x))$	24	0,11	13
$((1,000) + (((x) \cdot (x)) \cdot (x)) - ((2,000) \cdot (x)))$	32	0,11	14
$((1,000) + ((x) \cdot (((x) \cdot (x)) - (2,000))))$	69	0,09	29
$(((((x) \cdot (x)) \cdot (x)) - ((x) + (-1,000))) - (x))$	37	0,11	18
$((1,000) + (((x) \cdot (x)) - (2,000)) \cdot (x))$	29	0,09	16
$((\cos(-0,000)) - (x)) - ((x) - (((x) \cdot (x)) \cdot (x)))$	63	0,12	50
$((1,000) - (((x) + (x)) - ((x) \cdot (x)) \cdot (x)))$	73	0,11	48
$((((x) - ((3,000) \cdot (x))) + (((x) \cdot (x)) \cdot (x))) - (-1,000))$	58	0,13	32
$(((((x) \cdot (x)) \cdot (x)) - (((x) + (x)) - (\cos((x) - (x))))$	27	0,14	14
$((x) \cdot ((x) \cdot (x))) - ((-1,000) + ((x) + (x)))$	31	0,11	17
$((1,000) + (((x) \cdot (x)) \cdot (x)) - ((x) + (x)))$	35	0,11	23
$((((\cos(0,000)) - (x)) + ((x) \cdot ((x) \cdot (x))) - (x))$	44	0,12	22
Среднее	53,75	0,111	31,3

Таблица 3

Результаты анализа полученных решений

Показатели	Прямой метод	Метод Эйлера
Средний процент точных решений	35	62
Средний процент условно точных решений	25	25
Средний процент приближенных решений	40	13
Среднее количество поколений ГП	127	78
Среднее время работы, мин	475	321
Количество точных решений для наиболее сложной задачи	0	10
Количество приближенных решений для наиболее сложной задачи	100	30

Анализируя полученные данные, можно сделать вывод о том, что и прямой метод, и метод Эйлера имеют право на существование.

Прямой метод используется в случае поиска глобального экстремума, хотя в силу того что этот метод требует больше ресурсов и не отличается высокой надежностью (что можно объяснить накапливаемой вычислительной погрешностью при нахождении производных, а потом при численном определении значений функционала), его применение может быть обосновано только тем, что он не предполагает поиска производной более высокого порядка, как в методе Эйлера. Однако не каждую задачу можно решить таким методом, например в случае когда функционал неограничен сверху при поиске максимума (и снизу

при поиске минимума), решения задачи не существует (возможно, что это свойство данного метода может быть полезно для проверки ограниченности функционала).

Метод решения вариационной задачи с помощью уравнения Эйлера, т. е. перехода от вариационной задачи к краевой, показывает лучшие результаты и по времени работы, и по надежности, поэтому он представляется более полезным. Однако нужно отметить, что поставленная краевая задача не всегда имеет решение, а если и имеет, то это решение может быть и не единственным или вообще не являться решением вариационной задачи. Поэтому такие задачи и их решения должны быть подвергнуты дополнительному анализу.

Таким образом, предложен подход, основанный на алгоритме генетического программирования, позволяющий численно решать вариационную задачу в символьном виде, и разработана программа, реализующая этот подход, которая задает функционал в символьном виде и граничные условия и решает задачу с получением точной формулы, если она существует, или приближенного символьного выражения в случае когда решение нельзя выразить элементарными функциями. Возможно точное вычисление производных путем построения производной функции или приближенное – с помощью разностных схем. Процесс решения может быть реализован двумя методами. Решения, полученные прямым методом, являются глобальными экстремумами, но в случае, когда функционал неограничен, глобального экстремума не существует и, следовательно, задачу решить нельзя. При решении методом Эйлера время работы алгоритма значительно сокращается, однако получение решения является лишь необходимым условием существования минимума или максимума (возможно, локального). Поэтому предпочтение тому или иному методу решения должно отдаваться в зависимости от условий поставленной задачи, но в целом можно отметить, что метод Эйлера может использоваться чаще.

Областью применения данного алгоритма может выступать любая отрасль науки и техники, так или иначе связанная с решением вариационной задачи. Однако этот алгоритм нужно рассматривать как дополнительный инструмент в наборе уже имеющихся традиционных методов, таких как методы Рунге, Канторовича, Галеркина и др. Нужно также отметить, что

традиционные методы ориентируются на решение известных типов уравнений, что и является ограничением для их применения, поэтому эти методы и алгоритм генетического программирования нельзя сравнивать по точности или времени работы: очевидно, что традиционные методы будут быстрее, если решаемая задача удовлетворяет их теоретическим положениям. Тем не менее предложенный в данной статье алгоритм заслуживает внимания в тех случаях, когда стандартные методы не могут дать желаемого результата.

#### Библиографические ссылки

1. Эльсгольц Л. Э. Дифференциальные уравнения и вариационное исчисление. М. : Наука, 1969.
2. Koza J. R. Genetic Programming: On Programming Computer by Means of Natural Selection and Genetics. Cambridge, Mass. : MIT Press, 1992.
3. Бураков С. В., Семенкин Е. С. Решение задачи Коши для обыкновенных дифференциальных уравнений методом генетического программирования // Журн. Сиб. федер. ун-та. Серия «Математика и физика». 2011. Т. 4, № 1. С. 61–69.
4. Петровский И. Г. Лекции по теории обыкновенных дифференциальных уравнений. М. : Изд-во Моск. ун-та, 1984.
5. Самарский А. А. Теория разностных схем. М. : Наука, 1989.
6. Банди Б. Методы оптимизации (вводный курс). М. : Радио и связь, 1988.
7. Краснов М. Л., Макаренко Г. И., Киселев А. И. Вариационное исчисление. Задачи и упражнения. М. : Наука, 1973.

S. V. Burakov, E. S. Semenkin

#### ON SOLUTION OF VARIATIONAL PROBLEM WITH GENETIC PROGRAMMING TECHNIQUES

*The authors offer to use genetic programming techniques to solve variational problem. Two solving variants of finding solution are described. Features of the problem solution are considered. Results of numerical experiments are given.*

*Keywords: variational problem, ordinary differential equations, boundary problem, numeric methods, genetic programming algorithm.*

© Бураков С. В., Семенкин Е. С., 2011