

## ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ РЕШЕНИЯ КРАЕВЫХ ЗАДАЧ

### Введение

Генетические алгоритмы — это современный мощный инструмент решения сложных многопараметрических задач оптимизации. Идея генетического алгоритма (ГА) взята из живой природы и заключается в компьютерной организации эволюционного процесса создания, модификации и отбора лучших решений (в терминах ГА — особей) в целях появления новых, еще лучших вариантов решения задачи [1]. Каждая особь в ГА кодируется в виде строки (хромосома), состоящей из генов, в которых хранится информация о соответствующих параметрах решения. Оценкой качества закодированного в геноме решения служит значение функции приспособленности.

Генетические алгоритмы обладают рядом преимуществ: поиск решений осуществляется в них сразу из целого множества точек — популяции особей, используют только значения функции приспособленности и не требуют другой информации о поведении функции, относительно стойки к попаданию в локальные оптимумы, просты в реализации, могут использоваться для широкого класса задач и др. [2].

Первые публикации о применении принципов биологической эволюции к решению оптимизационных задач появились в конце 1960-х гг. А в 1975 году вышла известная работа Дж. Холланда «Адаптация в естественных и искусственных системах» [3], где собственно и был предложен первый генетический алгоритм. В дальнейшем в силу своей универсальности и высокой эффективности генетические алгоритмы получили широкое распространение для решения задач поиска и оптимизации в различных областях науки и техники [1, 2, 4, 5]. Недавно появились первые публикации об использовании ГА для решения некоторых краевых задач математической физики [6, 7].

В данной статье предлагается ГА для нахождения приближенного решения (в виде аналитического выражения) краевых задач для обыкновенных дифференциальных уравнений второго порядка и краевых задач для эллиптических уравнений, а также анализируются результаты численных экспериментов по решению тестовых задач и по исследованию влияния ряда параметров ГА на точность результатов и скорость сходимости алгоритма.

### Постановки задач

Исследования физических процессов в баллистике, теории упругости, теории колебаний, механике жидкостей и газов, динамике твердого тела, оптимальном управлении и других областях науки и техники часто приводят к краевым задачам для обыкновенных дифференциальных уравнений второго порядка. Эта задача ставится следующим образом: найти функцию  $u = u(x)$ , удовлетворяющую внутри отрезка  $[a, b]$  дифференциальному уравнению

$$f(x, u, u', u'') = 0, \quad (1)$$

а на концах отрезка — краевым условиям

$$g_1[u(a), u'(a)] = 0, \quad g_2[u(b), u'(b)] = 0. \quad (2)$$

В общем случае и уравнение (1), и краевые условия (2) могут быть нелинейными. Найти точное решение задачи (1), (2) (предполагается, что решение существует) в эле-

ментарных функциях удается только в отдельных случаях. Поэтому актуальной остается проблема нахождения приближенного решения, особенно нелинейных задач.

При решении краевой задачи (1), (2) приближенно-аналитическими методами поступают таким образом. Выбирают дважды непрерывно дифференцируемую на  $[a, b]$  функцию  $v = v(x; c_1, c_2, \dots, c_n)$ , которая при любых значениях входящих в нее параметров  $c_1, c_2, \dots, c_n$  точно удовлетворяет краевым условиям (2). Эту функцию подставляют в левую часть уравнения (1) и получают дифференциальную невязку

$$R(x; c_1, c_2, \dots, c_n) = f(x, v, v', v''), \quad (3)$$

являющуюся определенной характеристикой отклонения приближенного решения  $v$  от неизвестного точного решения  $u$  краевой задачи. Далее стараются найти такие значения параметров  $c_1, c_2, \dots, c_n$ , чтобы сделать функцию  $R$  наименее уклоняющейся от нуля в каком-то смысле в интересующей нас области [8]. Различные реализации этого требования приводят к различным методам решения краевой задачи: метод коллокации, Галеркина, наименьших квадратов (интегральный и дискретный), чебышевский метод и др. [8–10].

Исследования стационарных процессов разной физической природы (теплопроводность, диффузия, равновесие, задачи электростатики, магнитостатики, гидродинамики и т.д.) приводят к краевым задачам для дифференциальных уравнений с частными производными эллиптического типа [11]. В случае двух независимых переменных  $x$  и  $y$  эта задача формулируется следующим образом: найти функцию  $u = u(x, y)$  класса  $C^2(D) \cap C^1(\bar{D})$ , которая в области  $D \subset R^2$  удовлетворяет уравнению

$$Lu \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + p(x, y) \frac{\partial u}{\partial x} + q(x, y) \frac{\partial u}{\partial y} + r(x, y)u = g(x, y), \quad (4)$$

а на ее границе  $\Gamma$  — краевому условию

$$\alpha u + \beta \frac{\partial u}{\partial \bar{n}} = w(x, y), \quad (5)$$

где  $p, q, r, f, g$  — непрерывные функции,  $\bar{n}$  — внешняя нормаль к  $\Gamma$ ,  $\alpha$  и  $\beta$  — заданные числа, причем  $\alpha^2 + \beta^2 > 0$ .

Для решения краевой задачи (4), (5) можно использовать упомянутые выше приближенно-аналитические методы, подобно тому, как они применялись для приближенного решения обыкновенных дифференциальных уравнений. В этом случае подбирают функцию  $v = v(x, y; c_1, c_2, \dots, c_n)$ , которая при любых значениях параметров  $c_k$  удовлетворяет краевым условиям (5), и определяют постоянные  $c_k$  так, чтобы дифференциальная невязка  $R$

$$R(x, y; c_1, c_2, \dots, c_n) = Lv - g(x, y) \quad (6)$$

принимала возможно меньшее значение.

В итоге определение оптимальных значений параметров  $c_k$  решений задач (1), (2) и (4), (5) сводится в приближенно-аналитических методах к решению систем алгебраических уравнений. В случае нелинейных задач системы также будут нелинейными, и для их решения необходимо применение численных методов. Кроме того, вычисление матриц коэффициентов и координат векторов правых частей систем уравнений требует в ряде методов интегрирования по всему отрезку. Явное вычисление интегралов возможно только в отдельных несложных случаях, как правило, интегралы приходится вычислять приближенно.

В силу того, что традиционные приближенно-аналитические методы имеют слабые стороны, для нахождения решений краевых задач (1), (2) и (4), (5) предлагается другой подход. Он заключается в том, что задача определения наилучших значений параметров  $c_1, c_2, \dots, c_n$  рассматривается как оптимизационная задача поиска мини-

мума функции уклонения (чебышевского, среднеквадратического или др.) дифференциальной невязки  $R$  от нуля, и для решения оптимизационной задачи применяется генетический алгоритм. При таком подходе искомое решение представляется в виде хромосомы, генами которой являются значения параметров  $c_k$ . В этой работе используется чебышевское уклонение, определяемое как максимальное значение модуля невязки в рассматриваемой области [10, 12]. Кроме того, в численном эксперименте для сравнения приведены результаты решения задачи минимизации среднеквадратического уклонения.

### Генетический алгоритм для решения поставленных краевых задач

Схема стандартного ГА состоит из таких основных шагов.

1. Инициализация (создание начального поколения популяции).
2. Оценивание особей популяции по их значениям функции приспособленности.
3. Моделирование эволюционного процесса: отбор родителей для создания потомства, создание потомков отобранных родительских пар (скрещивание), мутация потомков, формирование нового поколения из расширенной популяции родителей и потомков (редукция и селекция).

4. Проверка критерия окончания алгоритма, и в случае его невыполнения повторение эволюционного процесса с шага 2.

Существуют различные варианты реализации стандартной схемы ГА, отличающиеся формой представления хромосом (двоичное и вещественное кодирование), операторами скрещивания, мутации и селекции, процедурами отбора родителей, критериями завершения алгоритма и т.д. Выбор того или иного варианта реализации стандартных составляющих при разработке ГА для решения конкретной задачи зависит, в частности, от особенностей предметной области, структуры данных и др.

Для решения поставленной задачи определения оптимальных значений параметров  $c_1, c_2, \dots, c_n$  нами разработан ГА, в котором используется представление генов хромосомы напрямую в виде вещественных чисел. Алгоритмы такого типа называются ГА с вещественным кодированием или непрерывными ГА. Они были созданы специально для поиска в непрерывных пространствах большой размерности. В большинстве случаев ГА с вещественным кодированием выполняют поиск оптимума лучше и быстрее, чем ГА с двоичным кодированием [13].

В предлагаемом алгоритме хромосома  $S_k$ , кодирующая  $k$ -ю особь, состоит из  $n$  генов  $S_k = (s_{k1}, s_{k2}, \dots, s_{kn})$ ,  $k = \overline{1, N}$ . Значение гена  $s_{ki}$  соответствует некоторому значению параметра  $c_i$  приближенного решения  $v$ , т.е. каждый ген отвечает за одну переменную. Эволюция популяции особей (хромосом) моделируется последовательностью поколений  $\{S_k(t)\}$ , где  $t$  — номер поколения ( $t = 0, 1, 2, \dots$ ).

Функция  $v$ , которая задается пользователем, должна точно удовлетворять краевым условиям при любых значениях параметров. Если краевые условия (2) линейны (при этом уравнение (1) может быть и нелинейным), то обычно полагают

$$v(x; c_1, \dots, c_n) = \varphi_0(x) + \sum_{k=1}^n c_k \varphi_k(x),$$

где  $\varphi_k$  — линейно независимые, дважды непрерывно дифференцируемые на  $[a, b]$  функции, удовлетворяющие однородным краевым условиям, а функция  $\varphi_0$  — неоднородным краевым условиям. В качестве функций  $\varphi_k$  обычно выбирают многочлены, тригонометрические функции и т.п.

Предлагаемый алгоритм имеет такие особенности реализации стандартной схемы ГА.

**1. Инициализация.** В начальное (нулевое) поколение  $\{S_k(0)\}$  включаются  $N$  особей  $S_1, S_2, \dots, S_N$ . Гены  $s_{k1}, s_{k2}, \dots, s_{kn}$  хромосомы  $S_k$  ( $k = \overline{1, N}$ ) — это случайные числа, генерируемые с помощью датчика случайных чисел из заданно-

го числового интервала (по умолчанию задается интервал  $[-1, 1]$ ). Границы этого интервала имеют значение лишь на шаге инициализации, в дальнейшей эволюции пространство поиска является в принципе неограниченным.

**2. Оценивание.** Каждая особь  $S_k$ ,  $k = \overline{1, N}$ , оценивается по ее значению функции приспособленности  $Fit$  (фитнесс-функции), которое вычисляется по формуле

$$Fit(S_k) = \Delta(S_k), \quad (7)$$

где  $\Delta(S_k)$  — чебышевское уклонение

$$\Delta(S_k) = \max_{1 \leq i \leq m} |R(x_i, y_i; s_{k1}, s_{k2}, \dots, s_{kn})|. \quad (8)$$

При этом для краевой задачи (1), (2) дифференциальная невязка  $R$  определяется по формуле (3), а для краевой задачи (4), (5) — по формуле (6). Чем ближе значение фитнес-функции (7) к нулю, тем более приспособленной является особь и тем ближе закодированные в ее генах значения параметров к оптимальным. Для вычисления уклонения (8) расчетная область покрывается сеткой, состоящей из  $m$  точек  $x_i$  в случае задачи (1), (2) или  $m$  точек  $(x_i, y_i)$  в случае задачи (4), (5), и затем в этих точках вычисляются значения производных дифференциальной невязки  $R$ . Значения производных можно вычислить двумя способами: численным (с определенным порядком малости) и аналитическим. Так как дифференцирование всегда выполняется по строгим правилам, то аналитическое вычисление производной не представляет особых трудностей. Кроме того, аналитически вычисленная производная может быть точнее. В программной реализации ГА используется аналитическое вычисление производных.

**3. Моделирование эволюционного процесса.** Отбор родителей для скрещивания осуществляется в соответствии с процедурой парного турнирного отбора. Из популяции случайным образом выбираются две особи, и та из них, значение фитнес-функции которой больше, помещается в промежуточный массив. После повторения этой операции  $N$  раз скрещиванию подвергаются все последовательные пары особей из промежуточного массива.

Скрещивание выполняется на основе смешанного BLX- $\alpha$  кроссовера с  $\alpha = 0,5$ . В этом случае пара родителей  $S_j = (s_{j1}, s_{j2}, \dots, s_{jn})$  и  $S_k = (s_{k1}, s_{k2}, \dots, s_{kn})$  производит одного потомка, у которого значением  $i$ -го гена является случайное число из интервала  $[z_1 - 0,5 \cdot d, z_2 + 0,5 \cdot d]$ , где  $z_1 = \min(s_{ji}, s_{ki})$ ,  $z_2 = \max(s_{ji}, s_{ki})$ ,  $d = z_2 - z_1$ ,  $i = \overline{1, n}$ . Особенность этого кроссовера в том, что значение гена потомка может лежать в области, выходящей за пределы значений родительских генов на величину  $0,5 \cdot d$ . Показано [13], что BLX- $\alpha$  кроссовер при  $\alpha = 0,5$  превосходит по эффективности большинство операторов скрещивания.

На шаге мутации случайным образом выбирается один потомок, у которого меняется значение одного случайного гена. Если мутации подвергается, например, ген  $s_{li}$  особи  $S_l$ ,  $l > N$ , то новое значение  $\bar{s}_{li}$  этого гена определяется по правилу:  $\bar{s}_{li} = s_{li} \pm \gamma \cdot \mu$ , где знаки  $+$  или  $-$  выбираются с равной вероятностью,  $\gamma = 0,5 \cdot d$  (величина интервала  $d$  находится через значения соответствующих родительских генов), а величина  $\mu$  вычисляется по формуле [2]

$$\mu = \sum_{i=1}^{\tau} \sigma(i) \cdot 2^{-i},$$

где  $\sigma(i) = 1$  с вероятностью  $1/\tau$ , в противном случае  $\sigma(i) = 0$ ,  $\tau$  — заданный параметр (по умолчанию  $\tau = 10$ ). Оператор мутации обеспечивает генетическое разнообразие популяции и таким образом помогает предотвратить застревание популяции в локальных минимумах.

На шаге формирования следующего поколения из расширенной популяции родителей и потомков с помощью редукции и селекции в новое поколение включаются только  $N$  особей с наименьшим значением функции приспособленности  $Fit$  (элитарный отбор).

**4. Критерий окончания.** Алгоритм завершает работу, если выполняется одно из двух условий: 1) значение фитнес-функции наилучшей особи в поколении меньше заданной величины  $\delta$ ; 2) достигнуто заданное максимальное число поколений  $t_{\max}$ .

При выполнении первого условия найденное решение будет близким к точному решению задачи, но при выполнении второго условия этого утверждать нельзя (см. обсуждение результатов задачи 2). Поэтому при программной реализации ГА целесообразно предусмотреть выдачу сообщения о номере выполненного терминального условия.

Предложенный алгоритм для решения краевых задач имеет ряд преимуществ: не требует модификации при переключении от линейных задач к нелинейным и от краевых задач для обыкновенных дифференциальных уравнений к краевым задачам для уравнений с частными производными эллиптического типа, находит решения нелинейных задач без применения каких-либо численных методов, прост в реализации, слабо связан с конкретной структурой задачи, что позволяет использовать его для решения других подобных задач.

#### Обсуждение результатов численных экспериментов

Для проверки эффективности применения разработанного ГА для нахождения решений краевых задач выполнены численные эксперименты по решению задач, для которых были известны их точные решения, а также проведено исследование влияния контролируемых параметров (настроек) ГА на результат и скорость сходимости алгоритма.

В качестве иллюстрации рассмотрим четыре краевые задачи и сравним решения, полученные с помощью ГА, с точными решениями этих задач.

**Задача 1.** Линейная краевая задача

$$x^2 u'' - u' - 2u - 1 = 0, \quad u(0) = u(1) = 0.$$

Точное решение  $u = x^2 - x$ .

**Задача 2.** Линейная краевая задача

$$u'' + u + x = 0, \quad u(0) = u(1) = 0.$$

Точное решение  $u = \sin(x)/\sin 1 - x$  [9].

**Задача 3.** Краевая задача для нелинейного дифференциального уравнения с линейными краевыми условиями

$$(u'')^2 - 4xu'' + 4u' = 0, \quad u(0) = 2, \quad u(1) = -4.$$

Точное решение  $u = 3x^2 - 9x + 2$ .

**Задача 4.** Краевая задача для уравнения эллиптического типа

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 2x(1-x), \quad 0 < x < 1, \quad 0 < y < 1;$$

$$u(0, y) = 0, \quad u(1, y) = 0, \quad 0 \leq y \leq 1;$$

$$u_y(x, 0) = 0, \quad u_y(x, 1) = 0, \quad 0 \leq x \leq 1.$$

Это задача о стационарном распределении температуры в однородной квадратной пластине, в середине которой действуют источники тепла интенсивностью  $2x(1-x)$ , если коэффициент внутренней теплопроводности равен 1, края  $x=0$  и  $x=1$  пластинки поддерживаются при нулевой температуре, а другие два края теплоизолированы.

Точное решение задачи  $u(x) = (2x^3 - x^4 - x)/6$  [11].

Для приближенного решения краевых задач 1–4 выбираем функцию  $v$ , которая при любых значениях входящих в нее параметров удовлетворяет заданным краевым условиям. Пусть число параметров  $n = 2$ . Тогда приближенное решение в задаче 1 и 2 ищем в виде  $v(x) = c_1x(1-x) + c_2x^2(1-x)$ , в задаче 3 —  $v(x) = 2 - 6x + c_1x(1-x) + c_2x^2(1-x)$  и в задаче 4 —  $v(x) = c_1x(1-x^2) + c_2x^3(1-x)$  (решение ищем в виде функции только переменной  $x$ , поскольку свободный член в дифференциальном уравнении этой задачи является функцией только переменной  $x$  и условия на краях  $x = 0$  и  $x = 1$  не зависят от переменной  $y$ ).

Для реализации предложенного алгоритма и решения поставленных краевых задач в системе программирования Borland Delphi была разработана соответствующая программа. В программе имеется две группы настроек: 1) настройки, определяющие текущую краевую задачу (само уравнение, границы области, число точек сетки, покрывающей заданную область); 2) настройки (параметры) собственно ГА (размер популяции, максимальное число поколений и др.). Значения параметров ГА определяются из потребности найти решение за наименьшее время или с наибольшей точностью и надежностью.

Численные эксперименты выполнялись при следующих настройках алгоритма: размер популяции  $N = 150$ , число генов  $n = 2$ , значение величины  $\delta = 0,000\,000\,001$  (первое терминальное условие), максимальное число поколений  $t_{\max} = 500$  (второе терминальное условие), расчетная область краевой задачи — отрезок  $[0, 1]$ , число точек сетки  $m = 101$ , все остальные настройки ГА по умолчанию. Поскольку в основе ГА лежит вероятностный подход, то приемлемое решение задачи можно получить только при многократных пусках алгоритма. В численном эксперименте для каждой задачи осуществлялось 15 пусков алгоритма, и в качестве оптимального выбиралось лучшее за все пуски решение.

Результаты решения краевых задач 1–4 с помощью предложенного алгоритма представлены в табл. 1–3. В табл. 1 приведены характеристики оптимальных решений всех четырех задач, в табл. 2 — результаты численного сравнения точного и приближенного решений задачи 4 в некоторых точках расчетной области и в табл. 3 — усредненные результаты по каждой задаче за все пуски алгоритма. Анализ данных этих таблиц показывает, что полученные решения задач 1, 3 и 4 хорошо согласуются с их точными решениями. В случае задачи 2 окончание алгоритма произошло в результате достижения максимального числа поколений популяции (терминальное условие 2). И хотя найденное решение не дает высокой точности приближения, оно, тем не менее, является наилучшим приближением заданного вида (кубический многочлен) к точному решению задачи 2, что подтверждается его полным совпадением с решением, найденным чебышевским методом [14]. Для получения более точного решения задачи 2 следует выбрать функцию  $v$  другого вида. Если решение искать, например, в виде  $v(x) = c_1x(1-x) + c_2x^2(1-x) + c_3x^3(1-x)$ , то погрешность этого решения (с оптимальными значениями  $c_1, c_2, c_3$ ) будет равна  $2,0 \cdot 10^{-4}$  [14], что на порядок меньше, чем погрешность приведенного в табл. 1 решения.

Таблица 1

Задача	Коэффициенты решения $v$		Значение функции $Fit$	Абсолютная погрешность
	$c_1$	$c_2$		
1	-0,99999999998	$2,034 \cdot 10^{-11}$	$4,898735 \cdot 10^{-11}$	$1,304234 \cdot 10^{-12}$
2	0,18374688185	0,16666666667	$3,4160430 \cdot 10^{-2}$	$2,998167 \cdot 10^{-3}$
3	-3,000000000004	$-2,803 \cdot 10^{-13}$	$8,585883 \cdot 10^{-11}$	$9,825474 \cdot 10^{-13}$
4	-0,166666666692	0,166666666706	$8,415785 \cdot 10^{-11}$	$7,256765 \cdot 10^{-12}$

Таблица 2

Значение $x$	Точное решение $u(x)$	Приближенное решение $v(x)$	Абсолютная погрешность	Невязка $R(x)$
0	0	0	0	—
0,2	-0,03093333333333	-0,030933333333794	$4,61 \cdot 10^{-12}$	$5,872 \cdot 10^{-11}$
0,4	-0,0496	-0,04960000000700	$7,0 \cdot 10^{-12}$	$7,968 \cdot 10^{-11}$
0,6	-0,0496	-0,04960000000633	$6,33 \cdot 10^{-12}$	$6,288 \cdot 10^{-11}$
0,8	-0,03093333333333	-0,030933333333660	$3,268 \cdot 10^{-12}$	$8,320 \cdot 10^{-11}$
1	0	0	0	—

Таблица 3

Задача	Среднее число поколений	Среднее значение функции $Fit$	Среднее значение абсолютной погрешности
1	118	$5,409103 \cdot 10^{-10}$	$7,826236 \cdot 10^{-11}$
2	500	$3,4160430 \cdot 10^{-2}$	$2,998167 \cdot 10^{-3}$
3	109	$6,811863 \cdot 10^{-10}$	$3,792329 \cdot 10^{-12}$
4	133	$6,863838 \cdot 10^{-10}$	$4,493041 \cdot 10^{-11}$

Поскольку ГА является стохастической процедурой, оценка его эффективности (надежности) осуществляется усреднением полученных результатов по многократным пускам (см. табл. 3). Среднее число поколений, необходимых для выполнения критерия окончания алгоритма, является показателем скорости сходимости ГА. Как показывают расчеты, для нахождения решения краевой задачи требуется в среднем 215 поколений.

Если при вычислении фитнес-функции вместо чебышевского уклонения (8) взять среднеквадратическое уклонение, вычисляемое по формуле

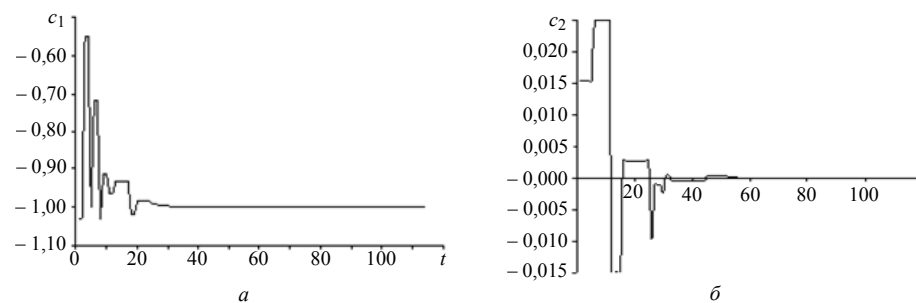
$$\Delta(S_k) = \sqrt{\frac{1}{m} \sum_{i=1}^m R^2(x_i, y_i; s_{k1}, s_{k2}, \dots, s_{kn})},$$

то получим результаты, приведенные в табл. 4. В этом случае для нахождения решения необходимо в среднем 209 поколений.

Таблица 4

Задача	Среднее число поколений	Среднее значение функции $Fit$	Среднее значение абсолютной погрешности
1	107	$6,120672 \cdot 10^{-10}$	$1,266050 \cdot 10^{-10}$
2	500	$2,095565 \cdot 10^{-2}$	$1,749968 \cdot 10^{-3}$
3	111	$5,471375 \cdot 10^{-10}$	$4,250811 \cdot 10^{-12}$
4	117	$5,525984 \cdot 10^{-10}$	$6,372099 \cdot 10^{-11}$

На рисунке наглядно представлен процесс оптимизации значений параметров  $c_1$  (а) и  $c_2$  (б) в ГА. Его можно разбить на две фазы: фазу грубой настройки, характеризующейся осцилляциями кривой; и фазу тонкой настройки [6, 7]. В первой фазе, на которую приходится в среднем 30 % поколений, преобладают крупномасштабные изменения, обеспечивающие широкую область поиска, а во второй фазе, занимающей 70 % поколений, происходит уточнение найденных «хороших» значений.



В эксперименте было выявлено, что процесс минимизации функции приспособленности тоже разбивается на две фазы длительностью в среднем 30 % и 70 % поколений соответственно. В первой фазе значение фитнес-функции наилучшей особи поколения уменьшается очень быстро, а во второй это уменьшение значительно замедляется [5].

Исследовалось влияние на результат и скорость сходимости ГА таких его контролируемых настроек, как размер популяции  $N$  и величина  $\delta$  из первого терминального условия. При изучении влияния размера популяции ее численность увеличивалась от  $N=50$  до  $N=400$  с шагом 50. При этом остальные настройки ГА оставались неизменными (значение  $\delta = 0,000\,0001$ ). Приведенные в табл. 5 результаты решения задачи 1 показывают, что при малых размерах популяции для сходимости ГА требуется большее число поколений, в то же время при использовании популяций большого размера увеличивается время выполнения алгоритма из-за существенного роста количества вычислений значений фитнес-функции. После  $N=300$  рост скорости сходимости ГА становится незначительным.

Аналогичное исследование проводилось относительно величины  $\delta$  из первого терминального условия (проверка выполнения неравенства  $Fit < \delta$ ). При этом значение  $\delta$  уменьшалось каждый раз на порядок, а остальные настройки ГА, в том числе и размер популяции  $N=150$ , оставались неизменными. Приведенные в табл. 6 результаты для задачи 3 показывают, что с уменьшением  $\delta$  растет число поколений, необходимых для сходимости алгоритма (поскольку увеличивается продолжительность процесса поиска более точного решения), при этом абсолютная погрешность постепенно уменьшается.

Таблица 5

Размер популяции	Среднее число поколений	Среднее значение функции $Fit$	Среднее значение абсолютной погрешности
50	139	$6,632945 \cdot 10^{-5}$	$1,225813 \cdot 10^{-5}$
100	114	$6,942499 \cdot 10^{-8}$	$8,439479 \cdot 10^{-9}$
150	91	$5,547410 \cdot 10^{-8}$	$8,057519 \cdot 10^{-9}$
200	86	$6,583819 \cdot 10^{-8}$	$7,803809 \cdot 10^{-9}$
250	82	$5,505646 \cdot 10^{-8}$	$8,814728 \cdot 10^{-9}$
300	78	$5,408939 \cdot 10^{-8}$	$8,144213 \cdot 10^{-9}$
350	77	$5,935696 \cdot 10^{-8}$	$7,490805 \cdot 10^{-9}$
400	76	$5,253956 \cdot 10^{-8}$	$7,237783 \cdot 10^{-9}$

Таблица 6

Значение $\delta$	Среднее число поколений	Среднее значение функции $Fit$	Среднее значение абсолютной погрешности
0,000 01	67	$5,712979 \cdot 10^{-6}$	$3,761177 \cdot 10^{-8}$
0,000 001	76	$5,791452 \cdot 10^{-7}$	$4,286590 \cdot 10^{-9}$
0,000 000 1	89	$5,788262 \cdot 10^{-8}$	$4,077400 \cdot 10^{-10}$
0,000 000 01	99	$6,302001 \cdot 10^{-9}$	$3,659419 \cdot 10^{-11}$
0,000 000 001	109	$6,811863 \cdot 10^{-10}$	$3,792329 \cdot 10^{-12}$
0,000 000 000 1	120	$5,903691 \cdot 10^{-11}$	$4,116463 \cdot 10^{-13}$
0,000 000 000 01	135	$6,120616 \cdot 10^{-12}$	$3,683692 \cdot 10^{-14}$

### Заключение

Предложенный в статье ГА позволяет находить в аналитической форме приближенные решения краевых задач для обыкновенных дифференциальных уравнений второго порядка и краевых задач для уравнений эллиптического типа. Результаты численного эксперимента показали, что полученные с его помощью решения тестовых задач хорошо согласуются с их точными решениями. Алгоритм не требует модификации при переключении от линейных задач к нелинейным,



находит решения нелинейных задач без применения численных методов, прост в реализации и может использоваться для решения других подобных задач. В статье проанализировано влияние некоторых контролируемых параметров ГА на точность решений и скорость сходимости алгоритма, что позволяет пользователю настраивать ГА в зависимости от потребности найти решение краевой задачи за наименьшее время или с наибольшей точностью и надежностью.

*Л.П. Вакал*

## ЗАСТОСУВАННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗАННЯ КРАЙОВИХ ЗАДАЧ

Запропоновано генетичний алгоритм для розв'язання крайових задач для звичайних диференціальних рівнянь другого порядку і крайових задач для еліптичних рівнянь. Наведено результати чисельних експериментів розв'язання лінійних і нелінійних задач. Знайдені розв'язки подаються в аналітичній формі. Досліджено вплив параметрів генетичного алгоритму на точність розв'язків і швидкість збіжності.

*L.P. Vakal*

## USING GENETIC ALGORITHM FOR SOLVING BOUNDARY VALUE PROBLEMS

A genetic algorithm for solving boundary value problems for second-order ordinary differential equations and boundary value problems for elliptic equations is proposed. Results of numerical experiments for solving linear and nonlinear problems are presented. The obtained solutions are presented in the analytical form. The influence of genetic algorithm parameters on the solutions accuracy and convergence rate is investigated.

1. Триус Ю.В., Триус В.Ю. Оптимізація багатоекстремальних функцій за допомогою гібридних методів у середовищі MATLAB R2007A // Вісник Черкаського університету. Прикладна математика. Інформатика. — 2010. — Вип. 172. — С. 104–122.
2. Панченко Т.В. Генетические алгоритмы. — Астрахань: Издательский дом «Астраханский университет», 2007. — 87 с.
3. Holland J.H. Adaptation in natural and artificial systems. — Ann Arbor: Univ. of Michigan Press, 1975. — 219 p.
4. Казаков В.Ю., Пейгин С.В., Тимченко С.В. Оптимизация по интегральному тепловому потоку траектории входа в атмосферу Земли затупленного тела // Прикладная механика и техническая физика. — 2000. — 41, № 4. — С. 112–123.
5. Вакал Л.П. Генетичні алгоритми для чебишовської апроксимації // Комп'ютерні засоби, мережі та системи. — 2013. — № 12. — С. 20–26.
6. Abu-Arquib Omar, Abo-Hammour Zaer, Momani Shaher, Shawagfeh Nabil. Solving singular two-point boundary value problems using continuous genetic algorithm. — <http://www.hindawi.com/journals/aaa/2012/205391>.
7. Abu-Arquib Omar, Abo-Hammour Zaer, Momani Shaher. Application of continuous genetic algorithm for nonlinear system of second-order boundary value problems // Appl. Math. Inf. Sci. — 2014. — 8, N 1. — P. 235–248.
8. Демидович Б.П., Марон И.А., Шувалова Э.З. Численные методы анализа. — М.: Наука, 1967. — 368 с.
9. Волков Е.А. Численные методы. — М.: Наука, 1987. — 248 с.
10. Вакал Л.П. Розв'язання крайових задач з використанням програмних засобів чебишовських наближень // Комп'ютерні засоби, мережі та системи. — 2010. — № 9. — С. 47–53.
11. Перестюк М.О., Маринець В.В., Рего В.Л. Збірник задач з математичної фізики. — Кам'янець-Подільський: Аксіома, 2012. — 249 с.
12. Вакал Л.П., Каленчук-Порханова А.О. Аналітична обробка даних на основі чебишовської апроксимації // Математичні машини і системи. — 2006. — № 2. — С. 15–24.
13. Herrera F., Lozano M., Verdegay J.L. Tackling real-coded genetic algorithms: operators and tools for the behaviour analysis // Artificial Intelligence Review. — 1998. — 12, N 4. — P. 265–319.
14. Каленчук-Порханова А.О., Вакал Л.П. Застосування найкращої чебишовської апроксимації для моделювання деяких фізичних процесів // Математичне та комп'ютерне моделювання. Серія технічні науки. — 2010. — № 4. — С. 111–118.

Получено 23.04.2015