

ПРИМЕНЕНИЕ АЛГОРИТМА КЛОНАЛЬНОГО ОТБОРА ДЛЯ РЕШЕНИЯ СИСТЕМ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

Abstract: In the given paper the features and characteristics of the clonal algorithm, intended for decision of the system algebraic equations are discussed. Clonal algorithm is very powerful method of combinatorial search; which makes decision on the base of sorting multiplicity variants of sequences from n -variables of them and represents more adaptive mechanisms than genetic algorithms when searching decisions.

Key words: clonal algorithm, genetic algorithm, combinatorial search, adaptive mechanisms.

Анотація: У даній роботі висвітлюються особливості і характерні риси клонального алгоритму, призначеного для рішення систем алгебраїчних рівнянь. Клональний алгоритм є могутнім методом комбінаторного пошуку, який будує своє рішення на основі перебору багаточисельних варіантів послідовностей із n -перемінних системи і отбору кращих з них і представляє більш адаптивні механізми, ніж генетичні алгоритми при пошуку рішень.

Ключові слова: клональний алгоритм, генетичний алгоритм, комбінаторний пошук, адаптивні механізми.

Анотация: В данной работе обсуждаются особенности и характерные черты клонального алгоритма, предназначенного для решения системы алгебраических уравнений. Клональный алгоритм является очень мощным методом комбинаторного поиска, который строит свое решение на основе перебора многочисленных вариантов последовательностей из n -переменных системы и отбора, лучших из них и предоставляет более адаптивные механизмы, чем генетические алгоритмы при поиске решений. **Ключевые слова:** клональный алгоритм, генетический алгоритм, комбинаторный поиск, адаптивные механизмы.

1. Введение

Иммунная система представляет большой интерес для специалистов в области информатики, так как она является примером мощных и гибких возможностей децентрализованной обработки информации. При этом иммунная система является примером биологической системы с превосходными адаптивными механизмами на локальном уровне и эмерджентных механизмов поведения на глобальном уровне. В настоящее время иммунные алгоритмы разрабатываются для решения множества задач в широкой области приложений, таких как задачи оптимизации, задачи распознавания образов и т.д. В соответствии с известными нам литературными данными, в настоящее время наибольшей популярностью из подходов, использующих иммунные алгоритмы, являются алгоритмы, применяющие механизмы отрицательного отбора, клонального отбора и модели иммунных сетей для решения задач автоматической генерации и настройки архитектуры нейронных сетей; в приложениях, ориентированных на прогнозы, защиты информации [1]. Мы провели ряд исследований по применению принципов и механизмов естественной иммунной системы, в частности теории клонального отбора. В настоящей работе мы описываем и используем алгоритм, основанный на механизмах клонального отбора.

2. Теория клонального отбора

Теория клональной селекции используется с тем, чтобы объяснить, как иммунная система “борется” против чужеродных антигенов [2]. Когда бактерия проникает в наш организм, она начинает размножаться и поражать своими токсинами клетки нашего организма. Была предложена одна из теорий, которая описывает, как иммунная система справляется с этими чужеродными антигенами. Те клетки, которые способны распознавать чужеродный антиген, размножаются асексуальным способом, пропорционально степени их распознавания: чем лучше распознавание

антигена, тем большее количество потомства (клонов) было сгенерировано. В течение процесса репродукции клетки отдельные клетки подвергаются мутации, которая позволяет им иметь более высокое соответствие (*аффинность*¹) к распознаваемому антигену: чем выше аффинность родительской клетки, тем в меньшей степени они подвергаются мутации, и наоборот. Обучение в иммунной системе обеспечивается увеличением относительного размера популяции и аффинности тех лимфоцитов, которые доказали свою ценность при распознавании представленного антигена. Основными иммунными механизмами при разработке алгоритма являются обработка определенного множества антител из набора клеток памяти, удаление антител с низкой аффинностью, созревание аффинности и повторный отбор клонов пропорционально их аффинности к антигенам [3, 4, 5].

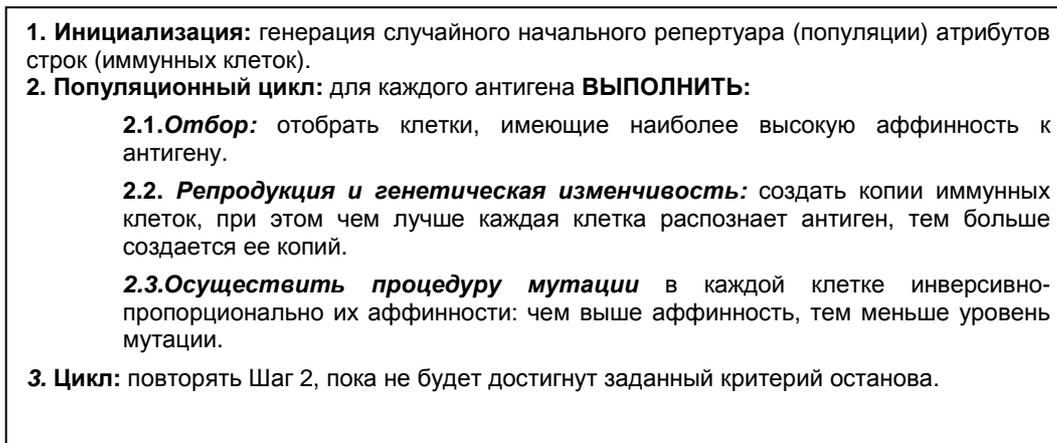


Рис. 1. Обобщенный алгоритм клонального отбора

Основываясь на данном механизме, в общем виде алгоритм клональной селекции представлен на рис. 1. Формально клональный алгоритм можно представить следующим образом [6]:

$$CLONALG = (Ab^0, Ag, L, N, n, \beta, d, \varepsilon), \quad (1)$$

где Ab^0 – исходная популяция антител; Ag – популяция антигенов; N – количество антител в популяции; L – длина рецептора антитела; n – количество антител, отбираемых для клонирования (с самой высокой аффинностью); β – множительный фактор, регулирующий количество клонов отобранных антител; d – количество антител, подлежащих замене новыми (т.е. имеющие самую низкую аффинность); ε – критерий останова. Если клональный алгоритм используется для решения задач оптимизации, а не распознавания, то популяция антигенов заменяется функцией, подлежащей оптимизации, – целевой функцией (f).

Несмотря на широту модификаций клонального алгоритма [7], все они имеют единый принцип работы: популяция антител подвергается некоторым преобразованиям, в результате которых индивидуумы увеличивают свою аффинность.

В клональном алгоритме, при решении той или иной задачи, можно использовать различные способы представления операторов, вычисление аффинности, параметров (сюда относят размер

¹ **Аффинность** – общий термин, относящейся к качеству элемента иммунной системы в отношении внешней среды, в работе которой он помещен, в данной работе – сила связывания между антигеном и антителом называется их

популяции, вероятности применения операторов гипермутации и замещения, количество итераций, условия остановки работы клонального алгоритма). Кроме того, они обладают универсальностью в решении различных видов задач, упрощают нахождение их решения, а также позволяют полностью автоматизировать процесс поиска оптимума [8].

Таким образом, клональный алгоритм работает с совокупностью “антител” (наборов решений) – популяцией, каждая из которых представляет возможное решение данной задачи. Каждое из антител оценивается мерой пригодности согласно тому, насколько “хорошо” соответствующее ей решение задачи. Наиболее подходящие антитела получают возможность “воспроизводить” потомство с помощью гипермутации. Это приводит к появлению новых антител, которые в процессе гипермутации улучшают своё соответствие целевой функции. Антитела с наименьшей аффинностью замещаются случайно сгенерированными антителами, равномерно распределенными по всей области определения целевой функции, что позволяет не замыкаться на локальных экстремумах, а исследовать всю целевую функцию. Таким образом, происходит итеративный процесс воспроизведения новых популяций из лучших представителей предыдущего поколения. Каждое новое поколение содержит более высокое соотношение характеристик, которыми обладают лучшие члены предыдущих поколений.

3. Постановка задачи исследования

В данной работе обсуждаются некоторые особенности применения и характерные черты клонального алгоритма, предназначенного для решения систем алгебраических уравнений, описаны некоторые отличительные особенности клонального алгоритма, приспособленного для решения систем алгебраических уравнений. Сюда входят задание целевой функции, а также процедура динамического перекодирования параметров.

4. Проблемы решения систем алгебраических уравнений

Система алгебраических уравнений представляет собой совокупность из m уравнений с n неизвестными и требуется найти последовательности из n чисел, которые одновременно удовлетворяют каждому из m уравнений [9]. Если все m уравнений линейны, то такая система называется системой линейных уравнений. Если одно или несколько уравнений нелинейны, система носит название нелинейной системы уравнений. В общем виде систему алгебраических уравнений можно представить следующим образом:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0; \\ f_2(x_1, x_2, \dots, x_n) = 0; \\ \dots\dots\dots \\ f_m(x_1, x_2, \dots, x_n) = 0. \end{cases} \quad (2)$$

В данной работе остановимся на решении систем, в которых число уравнений равно числу неизвестных, т.е. $m = n$. Систему линейных алгебраических уравнений часто представляют в виде произведения матрицы коэффициентов системы на столбец неизвестных:

аффинностью или степенью соответствия, комплементарности [1]. Является эквивалентом функции пригодности в эволюционных алгоритмах.

$$[A] \times (\bar{X}) = (\bar{B}), \quad (3)$$

где $[A]$ – матрица коэффициентов системы; \bar{X} – вектор-столбец переменных системы; \bar{B} – вектор-столбец свободных членов системы.

$$[A] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}; \quad (\bar{X}) = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}; \quad (\bar{B}) = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix}. \quad (4)$$

Существующие на сегодняшний день методы решения систем алгебраических уравнений не являются в достаточной степени универсальными, так как, согласно условиям их использования, система уравнений должна удовлетворять различного рода требованиям, ограничивающим область применения того или иного метода. Например, одними из методов решения системы уравнений (2) являются решение по правилу Крамера и методом Гаусса [10]. В случае больших систем ($n > 4$) решение по правилу Крамера оказывается очень громоздким, так как возникают трудности с нахождением определителей порядка больше 4. Метод Гаусса, хорошо применимый для систем линейных уравнений оказывается практически неприменим для решения нелинейных систем (за исключением тех случаев, когда нелинейную систему, посредством математических преобразований, например, заменой переменных, можно привести к линейному виду). Клональный алгоритм является очень мощным методом комбинаторного поиска, который строит свое решение на основе перебора многочисленных вариантов последовательностей из n переменных системы, называемых антителами, и отбора лучших из них путем оценки через прямую подстановку этих последовательностей во все уравнения системы. Таким образом, степень нелинейности системы не столь значительно сказывается на способности клонального алгоритма к ее решению, тем самым делая его наиболее универсальным методом решения систем алгебраических уравнений.

5. Предлагаемый метод решения

Создание клонального алгоритма начинается с построения целевой функции, которая содержит в себе информацию о целях задачи и их приоритетах. Для той или иной задачи цели могут выражаться по-разному. Так, целью задачи решения алгебраического уравнения вида

$$f(x) = 0 \quad (5)$$

является достижение баланса левой и правой частей. Целевую функцию для такого уравнения можно представить в виде

$$\Phi(x) = f(x) \rightarrow 0; \quad (6)$$

для клонального алгоритма более удобной будет запись

$$\Phi(x) = (f(x)) \rightarrow \min. \quad (7)$$

Тогда целевая функция для системы уравнений (1) представляет собой вектор, компонентами которого являются отдельные под целевые функции, стремящиеся к минимуму:

$$\Phi(x_1, x_2, \dots, x_n) = (\Phi_1(x_1, x_2, \dots, x_n), \Phi_2(x_1, x_2, \dots, x_n), \dots, \Phi_n(x_1, x_2, \dots, x_n)), \quad (8)$$

при $m = n$, где $\Phi_i(x_1, x_2, \dots, x_n) = (f_i(x_1, x_2, \dots, x_n)) (\rightarrow \min, i = \overline{1, n})$.

Следовательно, решение систем алгебраических уравнений сводится к решению задачи многоцелевой оптимизации с равными приоритетами целей. Множество G значений целевой функции Φ определяет множество вариантов решений системы

$$G = \{\bar{g}(g_1, g_2, \dots, g_n) \mid \bar{g}(g_1, g_2, \dots, g_n) = \Phi(x_1, x_2, \dots, x_n)\}, \quad (9)$$

среди которых клональный алгоритм должен найти наиболее предпочтительный вариант, т.е. вектор \bar{g}^* , все компоненты которого минимальны. Однако для нормальной работы клонального алгоритма необходимо иметь скалярное значение аффинности для каждого варианта решения. Для этого вводится функция аффинности, объединяющая все компоненты \bar{g} и дающая на выходе значение, которое воспринимается клональным алгоритмом как значение аффинности антител его популяции. В данной работе используется один из наиболее простых способов задания функции аффинности, выражающееся формулой

$$\varphi(\bar{g}) = \ln \left(\bar{g} \cdot \sum_{i=1}^n g_i \right) \text{ и} \quad (10)$$

$$\varphi(\bar{g}) \rightarrow \min.$$

Не менее важным элементом при создании клонального алгоритма является способ представления данных, с которыми он работает. В нашем случае в качестве данных выступают последовательности из n переменных системы в двоичном представлении. Каждая переменная x_i кодируется в виде строки бит размером c_i . Вследствие того, что все антитела имеют одинаковый, фиксированный размер, клональный алгоритм может работать на ограниченном множестве решений. Для того чтобы определить такое множество, в качестве параметров клонального алгоритма устанавливаются интервалы изменения для всех переменных системы с одинаковым шагом изменения внутри каждого интервала. Величина c_i определяет количество возможных вариантов значений переменной x_i в интервале $[a_i, b_i]$. В случае двоичного представления данных каждая переменная x_i может принимать в интервале $[a_i, b_i]$ 2^{c_i} значений. Тогда количество всех возможных вариантов решений системы (1) определится формулами:

$$K = \prod_{i=1}^n 2^{c_i} \text{ или } K = 2^{\sum_{i=1}^n c_i}, \quad (11)$$

где K – количество возможных вариантов решений системы алгебраических уравнений.

Из (11) видно, что увеличение параметра c_i вызывает увеличение K , а это может отрицательно сказываться на поисковой способности клонального алгоритма, так как в этом случае происходит значительное увеличение количества вариантов неверных решений, т.е. антител с низкой аффинностью, которые являются помехами в поисковом процессе. С другой стороны,

уменьшение значения величины c_i приводит к увеличению размера шага изменения переменной x_i в интервале $[a_i, b_i]$, что обуславливает появление "мертвых зон" в ландшафте решений, но которые могут содержать в себе лучшее решение. Для преодоления указанных трудностей в данной работе использован метод, называемый динамическим перекодированием параметров, который заключается в том, что при наступлении определенных условий в работе клонального алгоритма границы интервалов изменения всех переменных системы пересчитываются для новых, уменьшенных интервалов, и процесс продолжается до тех пор, пока не будет достигнуто нужное качество решения или не будет выполнено необходимое количество итераций. Границы интервалов в итерации i для i -й переменной системы рассчитываются по следующим формулам:

$$\begin{aligned} a_i^{it} &= x_{iB}^{it-1} - \frac{b_i^{it-1} - a_i^{it-1}}{2} \cdot V_i; \\ b_i^{it} &= x_{iB}^{it-1} + \frac{b_i^{it-1} - a_i^{it-1}}{2} \cdot V_i, \end{aligned} \quad (12)$$

где x_{iB}^{it-1} – значение переменной x_i для лучшего антитела, полученного в предыдущей итерации; b_i^{it-1}, a_i^{it-1} – границы интервала для i -й переменной в предыдущей итерации; V_i – функция сжатия интервала ($0 < V_i < 1$).

Условиями для пересчета границ является совпадение популяции в итерации i_{t-1} , т.е. ситуация, когда гены всех хромосом популяции в соответствующих позициях становятся одинаковыми. Характерные особенности некоторых систем алгебраических уравнений позволяют значительно ускорить процесс их решения при помощи клональных алгоритмов.

Рассмотрим, например, систему линейных уравнений вида

$$\begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45} & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & a_{i-1} & a_{ii} & a_{ii+1} & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & a_{n-1n-2} & a_{n-1n-1} & a_{n-1n} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & a_{nn-1} & a_{nn} \end{bmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \dots \\ x_i \\ \dots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ \dots \\ b_i \\ \dots \\ b_{n-1} \\ b_n \end{pmatrix}. \quad (13)$$

Как видно из выражения (13), ненулевые коэффициенты матрицы располагаются только на трех ее диагоналях. Каждое $i+1$ значение x может быть вычислено из i -го уравнения системы при известных x_i и x_{i-1} , т.е.:

$$x_{i+1} = f_i(x_i, x_{i-1}), \quad (14)$$

для первого уравнения

$$x_2 = f_1(x_1), \quad (15)$$

тогда после некоторых математических преобразований систему (13) можно представить в виде

$$\left\{ \begin{array}{l} x_2 = f_1(x_1); \\ x_3 = f_2(x_2, x_1); \\ x_4 = f_3(x_3, x_2); \\ \dots\dots\dots \\ x_{i+1} = f_i(x_i, x_{i-1}); \\ \dots\dots\dots \\ x_n = f_{n-1}(x_{n-1}, x_{n-2}); \\ f_n(x_n, x_{n-1}) = 0. \end{array} \right. \quad (16)$$

Последняя $(n - я)$ переменная системы вычисляется из $(n - 1)$ -го уравнения, а n -е уравнение представляет собой балансовое уравнение, в котором известны все переменные. На основании этого уравнения строится целевая функция, выражаемая формулой

$$\Phi = |f_n(x_n, x_{n-1})| \rightarrow \min, \quad (17)$$

при помощи которой можно непосредственно задавать значения пригодности хромосомам. Приведение системы (13) к виду (16) избавляет от необходимости включения в состав антитела полного набора переменных (x_1, \dots, x_n) . В данном случае антитело должно состоять только из одной переменной x_1 , что при прочих равных условиях (количество бит, используемое для кодирования переменных, фиксированный размер антител и т.д.) позволяет значительно сократить время работы алгоритма и расход памяти на содержание популяции антител.

Системы, подобные (13), часто используются для описания некоторых физических и физико-химических процессов. В частности, подобной системой пользуются в химической и нефтехимической промышленности для инженерных расчетов противоточных каскадов ректификационных колонн.

В качестве экспериментальных задач были выбраны две системы алгебраических уравнений:

– система линейных алгебраических уравнений:

$$\left\{ \begin{array}{l} x_1 - 0,7x_2 + 9x_3 + 3,6x_4 + 5,9x_5 + 3x_6 + 7x_7 = 22,9; \\ 2x_1 + 1,7x_2 - 5x_3 + 4,7x_4 + 5x_5 + 2,7x_6 - 3,7x_7 = 27; \\ 1,5x_1 + 2,6x_2 + 5,1x_3 + 7,6x_4 + 4,9x_5 + 8,3x_6 + 1,7x_7 = 37; \\ 1,8x_1 + 7,1x_2 + 3x_3 + 8,5x_4 - 5,9x_5 + x_6 + 7,8x_7 = 23,6; \\ 1,7x_1 + 6,7x_2 + 9,5x_3 - 1,6x_4 + 3,7x_5 - 9,5x_6 + 5,8x_7 = 45,1; \\ 7x_1 - 7,7x_2 + 5,6x_3 + 3,7x_4 + 2,5x_5 + 9x_6 + 7,7x_7 = 12,4; \\ 3x_1 + 9,7x_2 + 3,5x_3 - 7,6x_4 + 2,9x_5 + 3,1x_6 + 8,1x_7 = 34; \end{array} \right. \quad (18)$$

– нелинейная система алгебраических уравнений:

$$\left\{ \begin{array}{l} x_1 - x_2^2 + 3^{x_3 + x_4} - 5x_2x_5 - 2x_3 - 15 = 0; \\ 2(x_1 - x_2)^2 + 2x_2^3 + (x_3 - x_5)^3 + 15 = 0; \\ 3(x_2 - x_3)^2 - (x_2x_3 + 1)^2 + 3x_5^2 - 2 = 0; \\ 8x_1 - 7x_2^2 + (3x_2 - x_3x_4)^2 = 0; \\ x_1 \cdot 2^{x_2} + 5^{x_3} - 3x_4 + 3x_5^2 - 3 = 0. \end{array} \right. \quad (19)$$

Ввод:

n – количество переменных системы

$a(n), b(n)$ – границы интервалов изменения для каждой переменной системы

$c(n)$ – вектор представления переменных системы

$V(n)$ – вектор функций сжатия интервалов

ε – качество решения

l – количество итераций

loop until Decision_quality $\leq \varepsilon$ or $l = 0$

Vars = \emptyset

$i = 1$

loop until $i = n$

Vars _{i} = Variable_encoding_procedure(a_i, b_i, c_i) // Задание интервалов изменения для всех переменных

системы

$i = i + 1$

endloop

Population = Create_population(Vars) // Создание первой популяции антител

New_population = \emptyset // Оценка аффинности каждого антитела в популяции

Make_affinity(Population-1)

loop until Convergence // Условие: совпадение популяции

// Осуществление селекции антител с самой высокой аффинностью Клонирование антител с

//самой высокой аффинностью

Make_Clone_Antibody (Population_Best_antibody) // Гипермутация полученных лучших антител

Hypermutation(Best_Antibody) //Определение аффинности измененных клонов

Make_affinity_antibody_from (modified_best_antibody)

// Селекция измененных антител с самой высокой аффинностью

New_population = Select_antibody_from(modified_best_antibody) // Замена части старой популяции новой

Population-2 = Replace_antibody_from(New_population)

// Разнообразие репертуара: заменить подмножество антител с самой низкой аффинностью новыми // //

// индивидуумами

Neg_Select= Replace_worst_antibody_from (Population-2)

New_vars = \emptyset

$i = 1$

loop until $i = n$

New_vars _{i} = Variable_decoding_procedure(Best_antibody)// Процедура декодирования переменных системы

// Перекодирование параметров для новой итерации

$$a_i = \text{New_var } s_i - \frac{b_i - a_i}{2} \cdot V_i$$

$$b_i = \text{New_var } s_i + \frac{b_i - a_i}{2} \cdot V_i$$

$i = i + 1$

endloop

$I = I - 1$

endloop

Вывод: Best_antibody - лучшее решение

Рис. 2. Алгоритм динамического перекодирования

Эксперименты проводились с одинаковыми параметрами клонального алгоритма для обеих систем, приведенных ниже:

- начальный диапазон значений для всех переменных: [-50, 50];
- количество бит представления для всех переменных: $c = 5$;
- функция сжатия интервала для всех переменных представлена формулой

$$V_i = 1 - \frac{2^{s_i}}{2^{c_i}}, \quad (20)$$

где s_i – степень сжатия интервала ($0 < s_i < c_i$), $i = \overline{1, n}$; степень сжатия интервала для всех переменных: $s = 3$; размер популяции антител: 50; вероятность замещения: 0,9; вероятность гипермутации: 0,07.

Критерием качества решения выбрано значение функции аффинности, которое не должно превышать 0,0000005.

Эксперименты проводились на ЭВМ типа IBM PC AT 80486DX2-66. Решение системы линейных уравнений продолжалось в течение 1 мин. В результате были получены корни системы: $x_1 = 4,45331$; $x_2 = 3,12814$; $x_3 = 1,44209$; $x_4 = 1,31441$; $x_5 = 2,11429$; $x_6 = -0,42241$; $x_7 = -1,18305$. Решение нелинейной системы уравнений продолжалось в течение 42 секунд. В результате решения получены следующие корни: $x_1 = 0,70639$; $x_2 = -1,95692$; $x_3 = -0,92561$; $x_4 = 1,37341$; $x_5 = 1,49586$. Необходимо отметить, что корни (x_1, \dots, x_5) нелинейной системы не являются единственными на выбранном интервале. Поэтому для ее решения, на наш взгляд, целесообразно воспользоваться алгоритмом искусственной иммунной сети [11, 14], который использует дополнительно функцию антигенных взаимодействий и функцию взаимодействия внутри сети между антителами, которые добавляют эффект метадинамики, т.е. приток новых элементов и гибель нестимулируемых антител [12]. Этот механизм иммунной сети позволит одновременно определять несколько корней системы на заданном интервале.

6. Заключение

В данной работе рассмотрен случай использования алгоритма, использующего принципы клонального отбора и процесса созревания аффинности адаптивного иммунного ответа. Разработанный алгоритм предназначен для решения систем алгебраических уравнений. Следует отметить что, сравнивая клональный алгоритм и генетический алгоритм при решении рассматриваемого класса задач [13], важно, что основные шаги генетического алгоритма используются в клональном алгоритме. Однако, наряду с тем, что в генетическом алгоритме используется терминология, позаимствованная из генетики, на основе теории дарвинской эволюции в предложенном алгоритме используются понятия и механизмы, позаимствованные из теоретической иммунологии. Помимо этого, в данном алгоритме используется формализм, опирающийся на понятия взаимодействия антиген – антитело ($Ag - Ab$ – взаимодействия). В отличие от генетического алгоритма предложенный алгоритм осуществляет свой поиск при помощи механизмов соматической гипермутации, редактирования рецепторов и сбалансированного использования лучших решений с исследованием пространства поиска. Клональный алгоритм в целом может быть охарактеризован как кооперативный и вместе с тем как конкурентный подход, где отдельные антитела конкурируют за оптимизацию или распознавание антигенов, но целевая функция будет сотрудничать как ансамбль индивидуумов, представляющих конечное решение. Сравнивая наши ранние работы по решению систем алгебраических уравнений [13] с полученными

последними результатами с помощью клонального алгоритма следует отметить, что в то время, как генетический алгоритм имеет тенденцию концентрировать целую популяцию индивидуумов к лучшему кандидату решения, то клональный алгоритм позволяет получать разнообразное множество локальных решений, в том числе и глобальный оптимум. Что касается особенностей механизма работы клонального алгоритма, то следует заметить, что в данном алгоритме используется значение аффинности получаемого антитела (решения), для того, чтобы определить уровень мутации, который нужно применить к каждому члену популяции, в генетических же алгоритмах принимаемый уровень мутации и кроссинговера обычно игнорирует пригодность индивидуального решения (хромосомы). Таким образом, клональный алгоритм предоставляет более адаптивные механизмы, чем генетические алгоритмы при поиске решений.

СПИСОК ЛИТЕРАТУРЫ

1. De Castro L.N. & Timmis J. Artificial Immune Systems: A New Computational Intelligence Approach. – 2002. – 357 p.
2. Burnet F.M. A modification of Jerne's theory of antibody production using the concept of clonal selection. Australian Journal of Science 20. – 1957. – P. 67–69.
3. Ройт А. Основы иммунологии: Пер. с англ. – М.: Мир, 1991. – 328 с.
4. Ройт А., Бростофф, Мейл Д. Иммунология: Пер. с англ. – М.: Мир, 2000. – 582 с.
5. Вершигора А.Е. Общая иммунология: Учебное пособие. – К.: Выща школа, 1989. – 736 с.
6. Литвиненко В.И., Фефелов А.А., Горавский С.П. Объектно-ориентированная реализация алгоритма клональной селекции // Радиоелектроніка. Інформатика. Управління. – 2003. – № 9. – С. 81–88.
7. De Castro L.N., Von Zuben F.J. Artificial Immune Systems. – Part II: A Survey of Applications, Technical Report. – RT DCA 02/00, FEEC/UNICAMP. – Brazil, 2000. – 64 p.
8. De Castro L.N., Von Zuben F.J. The Clonal Selection Algorithm with Engineering Applications, submitted to GECCO'00. – 2000. – P. 36–37.
9. Бронштейн И.Н., Семендяев К.А. Справочник по математике для инженеров и учащихся втузов. – М.: Наука. Гл. ред. физ.-мат. лит., 1981. – 720 с.
10. Лихолетов И.И. Высшая математика, теория вероятностей и математическая статистика. – Минск: Высшая школа, 1976. – 720 с.
11. De Castro L.N., Von Zuben F.J. An Artificial Immune Network for Data Analysis // Data Mining: A Heuristic Approach, Hussein A. Abbass, Ruhul A. Sarker and Charles S. Newton (eds.) Idea Group Publishing. – USA, 2001. – Vol. XII. – P. 231–259.
12. Erelson A.S. Immune network theory // Immunological Reviews. – 1989. – Vol. 110. – P. 5–33.
13. Литвиненко В.И., Фефелов А.А., Ткачук А.А. Генетические алгоритмы в математическом моделировании // Вестник Запорожского университета. – 2001. – № 2. – С. 61–68.
14. Грицик В.В., Литвиненко В.И., Цмоць І.Г., Стех С.М. Теоретичні і прикладні проблеми застосування штучних імунних систем // Інформаційні технології і системи. – 2003. – Т. 6, №1–2. – С. 7–45.