

ОПТИМАЛЬНОЕ ПО ПОКАЗАТЕЛЯМ СВЯЗНОСТИ ОБЪЕДИНЕНИЕ СЕТЕЙ В УСЛОВИЯХ СТРУКТУРНЫХ И СТОИМОСТНЫХ ОГРАНИЧЕНИЙ

К. А. Нечунаева

Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия

УДК 004.023

Исследуются сети связи, моделируемые случайным графом. Разработан и реализован алгоритм, основанный на алгоритме клонирования. Проведено сравнение работы алгоритма клонирования, генетического алгоритма и алгоритма перебора.

Ключевые слова: объединение сетей, алгоритм клонирования, антитела, вероятность условной связности.

This paper deals with communication networks represented by random graphs. The main task of the work is integration of two networks with reliability optimization under structural and costs restrictions. A method based on Clonal Selection Algorithm (CSA) has been proposed for solving the problem. This method has been tested and compared with Genetic Algorithm (GA) and Enumeration Algorithm.

Key words: integration of networks, clonal selection algorithm, diameter constrained network reliability.

Введение. При системной интеграции информационных сетей важной задачей является объединение сетей на физическом уровне. При таком объединении необходимо обеспечить неухудшение качества связи, в частности выполнение ограничения на допустимое число коммутаций при установлении соединений.

В качестве модели сетей выбраны неориентированные графы с надежными вершинами и ненадежными ребрами. Это объясняется тем, что графы являются удобным средством для моделирования объектов, имеющих сложную структуру. Наиболее важным показателем надежности сети является ее связность, однако в данной работе выбирается более сильный показатель, который будем называть вероятностью условной связности. Иными словами, сеть связна, если между любыми ее двумя узлами существует путь, длина которого не превышает заданного числа d .

Проблема оптимального объединения графов по указанному выше показателю надежности – NP -полная комбинаторная проблема оптимизации, которая применяется для эффективного объединения двух или более реальных сетей. Проблема состоит в том, чтобы правильно расположить набор ребер между вершинами исходных графов, причем прокладывание ребер между сетями имеет определенную стоимость.

Главным инструментом решения данной задачи является алгоритм клонирования, основанный на принципах иммунной системы. Эффективные механизмы иммунной системы, такие как клонирование и мутация, делают искусственные иммунные системы полезным инструментом для решения комбинаторных проблем [1]. Кроме того, применяются некоторые алгоритмы теории графов для подсчета вероятности условной связности [2].

В данной работе приводятся некоторые основные понятия и сведения об известных методах. Приведен обзор работ, в которых рассматриваются различные подходы к изучаемой

проблеме. Представлена формальная постановка задачи, рассмотрен основной алгоритм для решения проблемы и проведено его сравнение с другими алгоритмами (генетическим и алгоритмом перебора).

1. Принципы искусственной иммунной системы. В последние годы резко возрос интерес к разработкам в области искусственных иммунных систем (AIS) и их приложениям. Среди многочисленных работ в этой новой области исследований следует отметить работы [1, 3-6]. Цель AIS – использование основных идей иммунологии для разработки систем, позволяющих найти оптимальные решения ряда практических задач. Основным алгоритмом, используемым в данной работе, является алгоритм клонирования, описанный в работе [1] и применявшийся в работе [6].

В случае, когда организм человека подвергается воздействию антигена (например, инфекции), его костный мозг начинает вырабатывать специальные клетки (В-лимфоциты), которые производят антитела (рис. 1). Каждая клетка секретирует только один тип антител, подходящий определенному антигену. Антиген стимулирует В-лимфоциты с антителами, наиболее подходящими для деления и образования новых В-лимфоцитов. В результате получается большое количество клонов одной клетки. Скорость клонирования пропорциональна пригодности антител. Часть новых клеток видоизменяется в плазматические клетки, которые очень активно производят антитела, остальные подвергаются мутации с целью получения наиболее подходящих антител для данного антигена. В-лимфоциты с наиболее подходящими антителами преобразуются в клетки памяти для запоминания нового антигена. В-клетки, не участвовавшие в клонировании, отмирают, в крови продолжают циркулировать клетки памяти до следующего воздействия антигена.

В алгоритме клонирования используются следующие обозначения: M – популяция; h – размер популяции; A_1, \dots, A_h , $A = \{x_1, \dots, x_{nm}\}$ – антитела ($x_i \in \{0, 1\}$).

Исследования, проведенные в работах [1, 3-6], показывают, что данный алгоритм является эволюционной стратегией, позволяющей решать такие сложные технические задачи, как распознавание образов, размещение предприятий и др.

2. Постановка задачи и некоторые вспомогательные алгоритмы. Для решения поставленной задачи потребуются следующие определения и понятия: $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ – ненаправленные графы; $G = (V, E)$ – объединенный граф; $V_1 = \{v_1^1, \dots, v_n^1\}$, $V_2 = \{v_1^2, \dots, v_m^2\}$, $V = V_1 \cup V_2$ – наборы вершин G_1 , G_2 и G соответственно; $E_1 = \{e_1^1, \dots, e_p^1\}$, $E_2 = \{e_1^2, \dots, e_q^2\}$, $E = E_1 \cup E_2$ – наборы ребер G_1 и G_2 ; $S = \{s_1, \dots, s_t\}$ – список ребер, которыми можно соединить вершины графов G_1 и G_2 ; c_1, \dots, c_t – стоимости ребер списка; n, m – число вершин G_1 и G_2 соответственно; x_e – состояние ребра e (1, когда ребро e исправно, и 0 в противном случае); $r_e = \Pr(x_e = 1)$ (или r_{ij}) – вероятность исправности ребра e (или e_{ij}); $q_e = \Pr(x_e = 0) = 1 - r_e$ – вероятность отказа ребра e ; $C(G)$ – стоимость ребер, использованных при построении G ; $D(G)$ – диаметр G , т. е. максимум кратчайших расстояний по

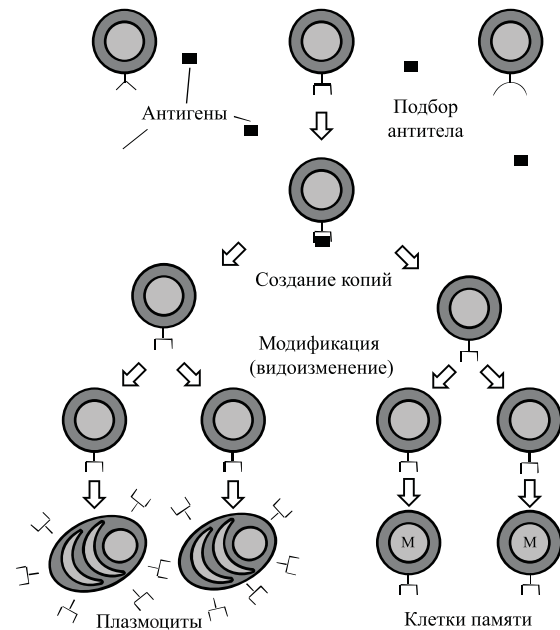


Рис. 1. Иммунная система человека

всем парам вершин графа. Понятие вероятности условной связности $R(G, d)$ определим как вероятность того, что между любыми двумя вершинами графа G существует путь, длина которого не превышает заданную длину d . Для вычисления этого показателя используем результаты работ [2, 7].

2.1. *Постановка задачи.* Пусть заданы неориентированные графы $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ с надежными вершинами $V_1 = \{v_1^1, \dots, v_n^1\}$, $V_2 = \{v_1^2, \dots, v_m^2\}$ и ненадежными ребрами $E_1 = \{e_1^1, \dots, e_p^1\}$, $E_2 = \{e_1^2, \dots, e_q^2\}$. Каждое ребро e присутствует с вероятностью r_e и отсутствует с вероятностью $q_e = 1 - r_e$. Диаметры этих графов не превышают заранее заданного числа d . Дан список ребер $S = \{s_1, \dots, s_t\}$, которые можно использовать для объединения G_1 и G_2 . Стоимости их различны и равны соответственно c_1, \dots, c_t . Требуется построить граф $G = (V, E)$ путем объединения двух данных графов с использованием ребер из списка, так чтобы диаметр полученного графа также не превышал d , а стоимость проложенных ребер не превышала C^* :

$$\begin{aligned} R(G, d) \rightarrow \max, \\ \text{cost}(G) < C^*. \end{aligned} \quad (1)$$

Будем оптимизировать вероятность условной связности. Для того чтобы решить эту задачу методом клонирования, перенесем теорию графов в термины и понятия искусственной иммунной системы.

Расстановку ребер между исходными графами зададим булевыми массивами размером $n \times m$, которые будем называть антителями. Единица в ячейке с номером $kn + l$, где $0 \leq k < m$, $0 < l \leq n$, означает, что было проложено ребро между $(k + 1)$ -й вершиной первого графа и l -й вершиной второго графа. Полученное в результате антителило показывает финальное размещение ребер между графами.

2.2. *Алгоритм подсчета вероятности условной связности.* Используем результаты работы [2]. Пусть имеется сеть $G = (V, E)$; нужно вычислить ее надежность $R(G, d)$ при заданном ограничении на диаметр d . Для любой пары узлов s, t определим $P_{st}(d)$ как набор путей между s и t , длина которых не превышает d . Обозначим через $E(p)$ случай, когда все ребра в p являются надежными. Вероятность случая $E(p)$ – вероятность обнаружения всех связей, составляющих путь p , в исправном состоянии; в соответствии с гипотезой о независимости отказов ребер

$$\Pr(E(p)) = \prod_{e \in p} r_e. \quad (2)$$

Тогда вынужденная диаметром мера надежности может быть выражена через функцию этих событий:

$$R(G, d) = \Pr\left(\bigcap_{s, t \in V} \left(\bigcup_{p \in P_{st}(d)} E(p)\right)\right). \quad (3)$$

Однако события $E(p)$ не являются ни независимыми друг от друга, ни непересекающимися, поэтому формула (3) непригодна для прямого вычисления меры надежности. Важным следствием является то, что для любого ребра $e \in E$, если оно не используется ни в одном пути между узлами, длина которых не превышает d , ребро не подходит для вычисления $R(G, d)$. В этом случае можно уменьшить граф, удаляя e и получая новую сеть $G - e$, которая будет иметь такую же надежность, что и первоначальная $R(G, d)$.

Введем некоторые определения. Пусть имеется граф $G = (V, E)$. Стиранием ребра e будем называть его удаление из набора ребер E сети G . Полученный граф обозначим $G - e$. Обнадеживанием ребра e будем называть признание его абсолютно надежным, т. е. $r_e = 1$. Полученный граф обозначим $G|e$.

Рассмотрим два возможных состояния произвольного ребра e , которое имеет некоторую вероятность отказа. Если ребро e нерабочее, то оно может быть удалено из сети; если оно работает, то может быть определено как надежное. Поскольку состояние ребра e не зависит от состояния других компонентов сети, имеем следующее выражение (более известное как *factoring theorem*):

$$R(G, d) = r_e R(G|e, d) + (1 - r_e) R(G - e, d). \quad (4)$$

Далее повторяем эту операцию для оставшихся ребер (таких, состояние которых еще не установлено) в сети G . Этот процесс соответствует алгоритму перечисления. Однако очень часто рекурсию можно остановить ранее; удаляя ребро e , можно получить, что независимо от состояния других ребер граф $G - e$ не имеет возможных путей между парой s, t , длина которых меньше либо равна d , и поэтому имеет надежность 0; также при абсолютно надежном ребре e может оказаться, что вершины будут связаны путями, составленными только из абсолютно надежных ребер, и сеть всегда будет работать независимо от отказа оставшихся ребер, тогда полученная надежность будет равна 1. В этих случаях продолжать рекурсию не имеет смысла.

Формула (4), дополненная предыдущими замечаниями, может использоваться как основа рекурсивного алгоритма. Для того чтобы проверить, следует ли продолжать рекурсию, используем следующие вспомогательные структуры. Предположим, что множество всех путей, длина которых не превышает d , между любыми парами вершин s и t найдено: $(P_{st}(d))$. Пусть $P(d)$ – объединение всех множеств $P_{st}(d)$ для всех пар вершин s и t . Введем следующие обозначения: np_{st} – количество всех путей длиной не более d между парой вершин s и t ; $links_p$ – количество ненадежных ребер e (таких, что $r_e < 1$) в пути $p \in P(d)$; $feasible_p$ имеет значение False, когда в пути p имеются неработающие ребра, и True в противном случае; $connected_{st}$ имеет значение True, когда между s и t существует абсолютно надежный путь длиной не более d , и False в противном случае; $connectedpairs$ – количество пар вершин, соединенных абсолютно надежными путями длиной не более d ; $P(e)$ – множество путей из $P(d)$, содержащих ребро e .

Следующий псевдокод считает $R(G, d)$:

Procedure FACTO(G, d)

Вход:

Граф $G = (V, E)$; ограничение на диаметр d ;

$P_{st}(d), P(e), np_{st}, links_p, feasible_p, connected_{st}, connectedpairs$

Выход: вероятность условной связности $R(G, d)$

Задание начальных условий:

$RContract = 0$ (временный результат для графа $G|e$)

$RDelete = 0$ (временный результат для графа $G - e$)

Произвольно выбираем ребро e : $0 < r_e < 1$.

“Обнадеживание”:

(Рассматриваются все надежные пути, содержащие ребро e)

for $p = (s, \dots, t) \in P(e)$ такого, что $feasible_p = \text{True}$ **do**

```

Сокращаем число ненадежных ребер в  $p$ :  $\text{links}_p = \text{links}_p - 1$ 
if  $\text{connected}_{st} = \text{False}$  и  $\text{links}_p = 0$  then
    ( $p$  – абсолютно надежный путь)
     $\text{connected}_{st} = \text{True}$ 
     $\text{connectedpairs} = \text{connectedpairs} + 1$ 
    if  $\text{connectedpairs} = n(n - 1)/2$  then
        (все вершины связаны абсолютно надежными путями)
         $\text{RContract} = 1$ 
        goto “Удаление”
    endif
endif
endfor
 $\text{RContract} = \text{ФАКТО}(G|e, d)$  (рекурсивный вызов процедуры)
“Удаление”:
(Вновь рассматриваются все надежные пути, содержащие ребро  $e$ )
for  $p = (s, \dots, t) \in P(e)$  такого, что  $\text{feasible}_p = \text{True}$  do
     $\text{feasible}_p = \text{False}$  (так как в  $p$  удаляется ребро  $e$ ,
    то путь становится непригодным)
     $\text{np}_{st} = \text{np}_{st} - 1$ 
    if  $\text{np}_{st} = 0$  then (граф несвязен)
         $\text{RDelete} = 0$ 
        goto “Итоговые вычисления”
    endif
endfor
 $\text{RDelete} = \text{ФАКТО}(G - e, d)$ 
“Итоговые вычисления”:
 $\text{R}(G, d) = r_e * \text{RContract} + (1 - r_e) * \text{RDelete}$ 
return  $\text{R}(G, d)$ 

```

2.3. *Функция родового сходства.* Функция родового сходства показывает, насколько данное антитело близко к оптимуму (будем считать его значение равным единице и стремиться к нему). Необходимо, чтобы полученное решение A имело наибольшее значение $\text{R}(G_A, d)$ вероятности условной связности, а стоимость проложенных ребер не превышала заданную C^* . Тогда функция родового сходства имеет вид

$$\text{Affinity}(A) = \begin{cases} 0, & \sum_{e \in S_A} \text{cost}(e) > C^*, \\ \text{R}(G_A, d), & \text{иначе,} \end{cases}$$

где G_A – сеть, объединенная из двух посредством антитела A ; S_A – список ребер, проложенных при построении G_A ; cost – функция стоимости.

2.4. *Функция вероятности выбора.* Используется также функция вероятности выбора, т. е. определяется пригодность антител для клонирования. Антителу с наибольшим родо-вым сходством будет соответствовать наибольшее количество клонов. Сначала вычисляем родо-вое сходство каждого антитела и в соответствии с полученными значениями нумеруем эти антитела. Рекурсивно вычислим функцию вероятности выбора (selection probability)

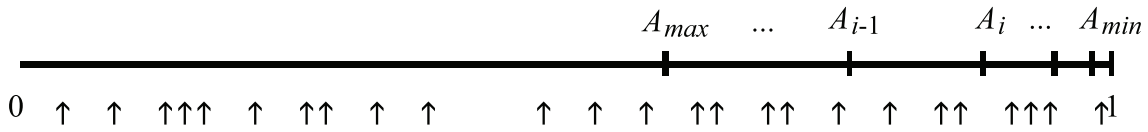


Рис. 2. Метод колеса рулетки

$$\text{Sel.Prob.}(A_i) = \begin{cases} \text{Affinity}(A_1) / \sum_{A_i \in M} \text{Affinity} A_i, & i = 1; \\ \text{Sel.Prob.}(A_{i-1}) + \text{Affinity}(A_i) / \sum_{A_i \in M} \text{Affinity} A_i, & 2 \leq i \leq h, \end{cases}$$

где M – популяция антител.

Для подсчета количества клонов у каждого антитела используем метод колеса рулетки, изложенный в работах [8, 9]. Расположим вероятности выбора антител в порядке возрастания на оси от 0 до 1, как показано на рис. 2 (A_{\max}, A_{\min} – антитела с максимальным и минимальным родовым сходством).

Генерируем h чисел от 0 до 1 (стрелки на рис. 1), тогда количество клонов данного антитела A_i будет соответствовать количеству случайных чисел, попавших в интервал $[A_{i-1}, A_i]$. Эта операция выполняется для того, чтобы наиболее пригодные антитела имели наибольшее количество клонов.

3. Описание основного алгоритма. Создадим популяцию антител, тогда алгоритм клонирования будет находить решение путем развития этих антител. Развитие основано на двух основных принципах позвоночной иммунной системы: выбор антител, пригодных для клонирования, и мутации. Предложенный алгоритм клонирования представлен ниже.

Создаем популяцию M антител размером h ;

for каждое поколение **do**

определяем родовое сходство (Affinity) антител;

вычисляем вероятность выбора (норма клонирования);

клонировуем (создаем копии антител);

for каждый созданный клон **do**

мутация инверсией (создаем новую последовательность (нп));

вычисляем родовое сходство нп;

if $\text{Affinity}(\text{нп}) > \text{Affinity}(\text{клон})$ **then**

клон = нп

else, do мутация перестановкой (создание нп);

вычисляем родовое сходство нп;

if $\text{Affinity}(\text{нп}) > \text{Affinity}(\text{клона})$ **then**

клон = нп

else, клон = клон;

антитело = клон;

исключаем худшее число антител $w\%$ в популяции

(w – параметр отношения отсеивания антител);

создаем новые антитела в таком же количестве ($w\%$ в популяции);

добавляем новые вместо отсеянных;

while критерий остановки = False.

Мутация инверсией. Пусть имеется антитело A , а i, j – два положения в последовательности A . Новое антитело получается из A инверсией последовательности отделов между i -м и j -м положениями в антителе.

Если родовое сходство видоизмененной последовательности (после мутации инверсией) больше, чем у оригинальной, то видоизмененная последовательность сохраняется вместо оригинальной. В противном случае последовательность вновь будет видоизменена в результате мутации перестановкой.

Мутация перестановкой. Пусть имеется антитело A , а i, j – произвольно взятые два положения в последовательности A . Новое антитело получено из A перестановкой отделов в положениях i и j .

Если родовое сходство видоизмененной последовательности (после мутации перестановкой) больше, чем у оригинальной, то видоизмененная последовательность сохраняется вместо оригинальной. Если алгоритм не может найти лучшее расположение после двух процедур мутации, то он сохраняет оригинальную последовательность (начальный клон).

Редактирование рецептора. После клонирования и процессов мутации устранена доля худших антител w % целого поколения. Произвольно созданные антитела добавляются в популяцию вместо устраненных. Этот механизм позволяет находить новые положения, которые соответствуют новым областям поиска в полном пространстве поиска. Исследование новых областей поиска может позволить алгоритму уходить дальше от локального оптимума.

Множество клонов принимается в качестве популяционного набора антител для следующего поколения после процессов клонирования, мутации и редактирования рецептора. Таким образом, клоны, которые были мутированы, определяются как антитела для следующего поколения. В следующем поколении клоны будут произведены от этих антител. Отсюда следует утверждение: антитело = клон в алгоритме.

На первых шагах алгоритма более вероятно найти лучшую последовательность, используя мутацию инверсией, поскольку алгоритм все еще далек от хороших решений и большее количество изменений может позволить найти лучшую расстановку ребер. На более поздних шагах у алгоритма будут хорошие решения. Возможность обнаружения лучших последовательностей с помощью большого количества изменений низка, так как возможны случаи потери хороших частичных расположений и может произойти отклонение от оптимума. Поэтому на более поздних шагах более эффективно выполнить относительно небольшое количество мутаций. В предложенном алгоритме эта эффективность обеспечена при использовании мутации перестановкой, когда мутация инверсией не дает лучшего решения.

4. Сравнение с другими алгоритмами. Для сравнения были выбраны генетический алгоритм и алгоритм перебора. Генетический алгоритм решает поставленную задачу в приемлемое время, используя похожий оператор мутации и оператор скрещивания, в то время как алгоритм перебора дает точный результат.

4.1. Генетический алгоритм. Решения, полученные с использованием алгоритма клонирования, сравниваются с решениями, найденными с помощью генетического алгоритма. Эти алгоритмы позволяют так же эффективно решать поставленную задачу и находить локальный оптимум.

Начальные данные: h – размер популяции, s % – доля выживших хромосом, p_m – вероятность мутации.

Схема алгоритма:

Создаем популяцию M хромосом размером h ;
for каждое поколение **do**:
 определяем родоное сходство (Affinity) хромосом;
 оставляем s % хромосом с наибольшим родовым сходством;
repeat:
 турнирной селекцией выбираем родителей;
 скрещиванием получаем новую последовательность (нп);
 мутация нп с вероятностью p_m ;
 нп добавляем в популяцию;
until размер популяции меньше h ;
while критерий останова = false.

Турнирная селекция. Из элементов популяции формируются два случайных непересекающихся подмножества, в каждом из них выбирается элемент с наибольшим значением функции родоного сходства. Это и будут родители.

Скрещивание. По хромосомам A_1, A_2 строится потомок A присваиванием каждой координате этого вектора с вероятностью 0,5 соответствующего значения одного из родителей. Если векторы A_1, A_2 совпали, например, по первой координате, то хромосома A “унаследует” это значение.

Мутация. Оператор мутации, применяемый к потомку A , полученному в процессе скрещивания, с заданной вероятностью p_m меняет значение каждой координаты на противоположное.

Сравнивая алгоритм клонирования (CSA) со стандартным генетическим алгоритмом (GA), следует отметить, что CSA может достигнуть разнообразного набора локальных оптимумов, в то время как GA имеет тенденцию полностью фокусироваться на одном локальном оптимальном решении.

4.2. *Алгоритм перебора.* Также сравним алгоритм клонирования с алгоритмом перебора. Так как алгоритм перебора находит точное оптимальное решение задачи, то будем использовать его в качестве проверки точности полученного с помощью алгоритма клонирования локального оптимума. Однако в случае большого количества вершин время нахождения точного решения для алгоритма перебора очень велико, поэтому данный алгоритм неэффективен при решении поставленной задачи.

5. Реализация. Рассмотренные алгоритмы были реализованы в среде C++ Builder и протестированы на различных графах небольшой размерности. В таблице представлены результаты тестирования алгоритма клонирования, генетического алгоритма и алгоритма

Сравнение работы алгоритмов

Алгоритм	Циклы 3, 4		Решетки 3×3 и 2×3	
	Время, с	Вероятность	Время, с	Вероятность
Вероятность мутации 0,1				
Клонирования	1,394	0,25	$3,90 \cdot 10^2$	0,013130188
Генетический	6,259	0,0625	$1,03 \cdot 10^3$	0,002750397
Перебора	1,204	0,25	–	–
Вероятность мутации 0,001				
Клонирования	1,394	0,25	$3,90 \cdot 10^2$	0,013130188
Генетический	$6,299 \cdot 10^{-2}$	0,03125	$1,20 \cdot 10^2$	0,001342773
Перебора	1,204	0,25	–	–

перебора на графах-решетках и циклических графах. В некоторых случаях решение, полученное с помощью алгоритма перебора, опущено вследствие очень большого времени работы.

Из данных, приведенных в таблице, следует, что алгоритм клонирования эффективнее, чем генетический, так как дает более точный результат.

Заключение. В данной работе приведен алгоритм клонирования, используемый для объединения сетей и оптимизации показателя вероятности условной связности с ограничениями на структуру (диаметр конечного графа) и стоимость проложенных ребер, что приближает данную задачу к реальным сетям. Разработаны вспомогательные функции и методы.

Реализация алгоритма клонирования проведена в среде C++ Builder, также для сравнения были реализованы генетический алгоритм и алгоритм перечисления, проведены эксперименты.

Методы AIS широко используются, так как разработка и изучение принципов иммунной системы позволяют решать различные проблемы с достаточно хорошим приближением к оптимуму. В дальнейшем планируется продолжить изучение методов и алгоритмов, основанных на природных принципах.

Список литературы

1. ISHIDA Y. The immune system as a self-identification process: a survey and a proposal // Proc. of the ICMAS Intern. workshop on immunity-based systems, Kyoto (Japan), 1996. Kyoto: MIT Press, 1996. P. 2-12.
2. CANCELA H., PETINGI L. Diameter constrained network reliability: exact evaluation by factorization and bounds // Proc. of the Intern. conf. on industrial logistics (ICIL'2001), Okinawa (Japan), 9-12 July 2001. S. l.: Intern. centre innovation industrial logistics, 2001. P. 359-366.
3. HUNT J. E., COOKE D. E. Learning using an artificial immune system // J. Network Comput. Appl. 1996. V. 19, iss. 2. Special issue on intelligent systems: design and applications. Pt 2. P. 189-212.
4. DASGUPTA D. Artificial immune systems and their applications. N.Y.: Springer-Verlag. 1999.
5. HOFMEYR S. A., FORREST S. Immunity by design: An artificial immune system // Proc. of the Intern. conf. on genetic and evolutionary computation (GECCO'99), Orlando (USA), 1999. S. l.: Morgan Kaufmann, 1999. P. 1289-1296.
6. ULUTAS B. H., ISLIER A. A. Parameter setting for clonal selection algorithm in facility layout problems // Proc. of the ICCSA-2007, LNCS 4705, Pt 1. P. 886-899.
7. НЕЧУНАЕВА К. А. Расчет и оптимизация показателей связности сети в условиях стоимостных и структурных ограничений: Диплом. работа бакалавра. Новосибирск, 2007.
8. DE CASTRO L. N., VON ZUBEN F. J. The clonal selection algorithm with engineering applications in workshop // Proc. of the Intern. conf. on genetic and evolutionary computation (GECCO) and workshop on artificial immune systems and their applications, Las Vegas (USA), 2000. S. l.: Morgan Kaufmann, 2000. P. 36-37.
9. GEN M. Genetic algorithms and engineering design / M. Gen, R. Cheng. N. Y.: John Wiley and Sons Inc., 1997.

Нечунаева Ксения Александровна – асп. Института вычислительной математики и математической геофизики СО РАН; e-mail: ksu.nech@rav.sccc.ru

Дата поступления – 01.04.10