

УДК 004

Е. Е. ФЕДОРОВ

*Донецкая академия автомобильного транспорта, Украина*

## РАЗРАБОТКА НОВОЙ МОДЕЛИ Т-КЛЕТОК ДЛЯ РЕШЕНИЯ ЗАДАЧ УСЛОВНОЙ ОПТИМИЗАЦИИ

Для создания интеллектуальной системы управления и прогноза в статье был предложен метаэвристический метод, основанный на принципах модели Т-клеток иммунных систем и неоднородной мутации, которая используется в генетическом алгоритме. Предложенная новая модель Т-клеток позволяет на ранних стадиях работы алгоритма исследовать все пространство поиска, а на заключительных стадиях сделать поиск направленным. Кроме того, эта модель позволяет использовать только вещественное представление клеток (потенциальных решений), что позволяет уменьшить вычислительную сложность. Разработанный метод был сравнен с аналогичными методами условной оптимизации числовых функций.

**Ключевые слова:** интеллектуальная система управления и прогноза, новая модель Т-клеток, иммунная система, генетический алгоритм, условная оптимизация числовых функций, неоднородная мутация, метаэвристика.

### Введение

*Постановка проблемы.* В настоящее время актуальной является разработка интеллектуальных компьютерных систем, предназначенных для прогноза и управления различными динамическими объектами (например, в авиационной промышленности). Для таких систем важнейшими задачами являются поиск оптимального управляющего воздействия и идентификация параметров, решение которых осуществляется методами условной оптимизации числовых функций.

*Анализ исследований.* Современные методы условной оптимизации числовых функций, как и методы комбинаторной оптимизации [1], все чаще используют подходы искусственного интеллекта (более узко – вычислительного интеллекта [2]). При этом важную роль играют два направления – эволюционные вычисления (чаще всего это генетический алгоритм, использующий штрафные функции [3]) и иммунные системы (чаще всего эти системы основаны на клональном отборе [4,5] или механизме Т-клеток [6]), а также их гибрид, представленный в работах [7 - 9]. Недостатком штрафных функций, используемых в генетическом алгоритме, является требование настройки параметров, что требует значительного числа экспериментов и зависит от предметной области. Первым недостатком иммунных метаэвристик [2 - 4] является то, что они требуют бинарного представления клеток (потенциальных решений) популяции, что снижает вычислительную эффективность. Вторым недостатком заключается в

отсутствии зависимости мутации от номера итерации, что затрудняет сужение пространства поиска на заключительных этапах работы метаэвристического алгоритма.

*Цель и задачи исследования.* Для повышения эффективности решения задач условной оптимизации числовых функций необходимо разработать метаэвристический метод, базирующийся на принципах модели Т-клеток иммунной системы и неоднородной мутации генетического алгоритма.

В статье:

- исследуется классическая модель Т-клеток;
- предлагается новая модель Т-клеток;
- проводится сравнительный анализ метаэвристических методов.

### 1. Основные положения модели Т-клеток

Модель Т-клеток (T-cell model) предложена Арагон (Aragon), Эсквевел (Esquivel) и Коэлло Коэлло (Coello Coello) [4], и основана на механизме Т-клеток. Ее отличительной особенностью является использование на каждой итерации трех субпопуляций клеток – новых (девственных) клеток, исполнительных клеток и клеток памяти, а также возможность использования функции цели с ограничениями.

Субпопуляция новых клеток создается на каждой итерации случайным образом. Субпопуляция исполнительных клеток на каждой итерации создается следующим образом: субпопуляция новых клеток разбивается на подмножество, содержащее

клетки, для которых сумма значений всех ограничивающих функций меньше адаптивного порога, и подмножество, содержащее клетки, для которых сумма значений всех ограничивающих функций превышает адаптивный порог; первое подмножество упорядочивается по функции цели, второе подмножество упорядочивается по сумме значений всех ограничивающих функций; определенное количество первых клеток из первого подмножества и определенное количество первых клеток из второго подмножества образуют субпопуляцию исполнительных клеток. Затем субпопуляция исполнительных клеток подвергается мутации. Субпопуляция клеток памяти создается (на первой итерации) или модифицируется (на последующих итерациях) следующим образом: субпопуляция мутировавших исполнительных клеток разбивается на подмножество, содержащее клетки, для которых сумма значений всех ограничивающих функций меньше адаптивного порога, и подмножество, содержащее клетки, для которых сумма значений всех ограничивающих функций превышает адаптивный порог; первое подмножество упорядочивается по функции цели, второе подмножество упорядочивается по сумме значений всех ограничивающих функций; определенное количество первых клеток объединения этих двух подмножеств на первой итерации образует субпопуляцию клеток памяти, а на последующих итерациях заменяют вторую (худшую) половину клеток памяти. Затем субпопуляция клеток памяти подвергается мутации. После чего субпопуляция мутировавших клеток памяти упорядочивается следующим образом: субпопуляция клеток памяти разбивается на подмножество, содержащее клетки, для которых сумма значений всех ограничивающих функций меньше фиксированного порога, и подмножество, содержащее клетки, для которых сумма значений всех ограничивающих функций превышает фиксированный порог; первое подмножество упорядочивается по функции цели, второе подмножество упорядочивается по сумме значений всех ограничивающих функций; объединение этих подмножеств образует модифицированную субпопуляцию клеток памяти. Результатом работы является лучшая (первая) клетка памяти.

Субпопуляция новых клеток больше субпопуляции исполнительных клеток (обычно в 2,5 раза), а субпопуляция исполнительных клеток больше субпопуляции клеток памяти (обычно в 2 раза). Для новых клеток не используется мутация, для исполнительных клеток и клеток памяти используются разные типы мутации. Мутация для исполнительных клеток работает с бинарным представлением клеток, а мутация для клеток памяти работает с вещественным представлением клеток, поэтому в

классической модели Т-клеток для субпопуляции новых клеток и исполнительных клеток приходится использовать оба представления, что повышает вычислительную сложность метода.

## 2. Алгоритм классической модели Т-клеток

Алгоритм классической модели Т-клеток, использующий бинарное и вещественное представление клеток, состоит из следующих шагов:

### 1. Инициализация.

1.1. Задание вероятности мутации исполнительных клеток  $p^E$ , параметра для мутации клетки памяти  $\alpha$ , причем  $\alpha$  – натуральное число.

1.2. Задание максимального числа итераций  $N$ , длины клетки  $M$ , количества бит для каждой компоненты клетки  $E$ , размера субпопуляции новых клеток  $L_V$ , количества отбираемых новых клеток с учетом ограничений  $L1_V$ , количества отбираемых новых клеток без учета ограничений  $L2_V$ , количества мутаций каждой исполнительный клетки  $N_E$ , размера субпопуляции клеток памяти  $L_M$ , количества мутаций каждой клетки памяти  $N_M$ , причем  $L_V < L_M$ .

1.3. Задание минимальных и максимальных значений для клетки  $x_j^{\min}, x_j^{\max}, j \in \overline{1, M}$ .

1.4. Задание функции стоимости (функция цели) и ограничений:

$$F(x) \rightarrow \min_x,$$

$$h_z(x) = 0, z \in \overline{1, Z_1},$$

$$g_z(x) \leq 0, z \in \overline{1, Z_2},$$

где  $x$  – клетка (вещественный вектор),

$Z_1$  – количество ограничений-равенств,

$Z_2$  – количество ограничений-неравенств.

1.5. Создание случайным образом лучшей клетки

$$x^* = (x_1^*, \dots, x_M^*),$$

$$x_j^* = x_j^{\min} + (x_j^{\max} - x_j^{\min}) \text{rand}(),$$

где  $\text{rand}()$  – функция, возвращающая равномерно распределенное случайное число в диапазоне  $[0,1]$ .

2. Номер итерации  $n = 1$ .

3. Создание субпопуляции новых клеток  $P^V$ .

3.1. Номер клетки  $k = 1, P^V = \emptyset$ .

3.2. Создание случайного бинарного вектора

$$b_k = (b_{k1}, \dots, b_{k, M \cdot E}),$$

$$b_{kq} = \begin{cases} 1, & \text{rand}() < 0,5; \\ 0, & \text{rand}() \geq 0,5. \end{cases}$$

3.3. Преобразование бинарного вектора в вещественный

$$x_k = (x_{k1}, \dots, x_{kM}),$$

$$x_{kj} = x_j^{\min} + (x_j^{\max} - x_j^{\min}) \frac{\sum_{e=1}^E (2^{E-e} \cdot b_{k,(j-1)E+E+1-e})}{2^E - 1}$$

3.4. Суммируются значения всех ограничивающих функций для  $i$ -й клетки

$$s_k = \sum_{z=1}^{Z_1} |h_z(x_k)| + \sum_{z=1}^{Z_2} \max\{0, g_z(x_k)\}.$$

3.5. Если  $(b_k, x_k, s_k) \notin P^V$ , то  $P^V = P^V \cup \{(b_k, x_k, s_k)\}$ ,  $k = k + 1$ .

3.6. Если  $k \leq L_V$ , то переход на шаг 3.2.

4. Вычисление значения адаптивного порога  $\Delta_V$  для субпопуляции  $P^V$

$$\Delta_V = \frac{1}{L_V} \sum_{k=1}^{L_V} s_k.$$

Если  $\Delta_V < 0,0001$ , то  $\Delta_V = 0,1$ .

5. Создание субпопуляции исполнительных клеток  $P^E$ .

5.1. Разбиение субпопуляции новых клеток  $P^V$  на подмножество  $P1^V = \{(b1_k, x1_k, s1_k)\}$ , содержащее клетки, для которых  $s1_k < \Delta_V$ , и подмножество  $P2^V = \{(b2_k, x2_k, s2_k)\}$ , содержащее клетки, для которых  $s2_k \geq \Delta_V$ .

5.2. Упорядочение подмножества  $P1^V$  по функции цели, т.е.  $F(x1_k) < F(x1_{k+1})$ .

5.3. Упорядочение множества  $P2^V$  по сумме значений всех ограничивающих функций, т.е.  $s2_k < s2_{k+1}$ .

5.4.  $L1_V$  первых клеток из упорядоченного множества  $P1^V$  и  $L2_V$  первых клеток из упорядоченного множества  $P2^V$  образует субпопуляцию исполнительных клеток  $P^E = \{(b_i, x_i, s_i)\}$  мощностью  $L_E = L1_V + L2_V$ , причем первыми идут клетки из множества  $P1^V$ .

6. Модификация субпопуляции исполнительных клеток  $P^E$  на основе мутации.

6.1. Номер клетки  $i = 1$ .

6.2. Номер мутации  $l = 1$ .

6.3. Вычисление максимального значения всех ограничивающих функций

$$s^{\max} = \max\{|h_1(x_i)|, \dots, |h_{Z_1}(x_i)|, \max\{0, g_1(x_i)\}, \dots, \max\{0, g_{Z_2}(x_i)\}\}.$$

6.4. Выполнить мутацию.

6.4.1. Создание бинарного вектора

$$\hat{b}_i = (\hat{b}_{i1}, \dots, \hat{b}_{i,M \cdot E}).$$

Если  $s^{\max} \leq \frac{s_i}{Z_1 + Z_2}$ , то

$r = \text{rand}()$ ,

$$\hat{b}_{iq} = \begin{cases} 1, & (r < p^E \wedge b_{iq} = 0) \vee (r \geq p^E \wedge b_{iq} = 1); \\ 0, & (r < p^E \wedge b_{iq} = 1) \vee (r \geq p^E \wedge b_{iq} = 0); \end{cases}$$

$q \in \overline{1, M \cdot E}$ .

Если  $s^{\max} > \frac{s_i}{Z_1 + Z_2}$ , то

$\hat{b}_i = b_i$ ,

$j = \text{round}(1 + (M - 1)\text{rand}())$ ,

$r = \text{rand}()$ ,

$$\hat{b}_{iq} = \begin{cases} 1, & (r < p^E \wedge b_{iq} = 0) \vee (r \geq p^E \wedge b_{iq} = 1); \\ 0, & (r < p^E \wedge b_{iq} = 1) \vee (r \geq p^E \wedge b_{iq} = 0); \end{cases}$$

$q \in \overline{(j-1)E + 1, jE}$ ,

где  $\text{round}()$  – функция, округляющая число до ближайшего целого.

6.4.2. Преобразование бинарного вектора в вещественный

$$\hat{x}_i = (\hat{x}_{i1}, \dots, \hat{x}_{iM}),$$

$$\hat{x}_{ij} = x_j^{\min} + (x_j^{\max} - x_j^{\min}) \frac{\sum_{e=1}^E (2^{E-e} \cdot \hat{b}_{i,(j-1)E+E+1-e})}{2^E - 1}.$$

6.5. Суммируются значения всех ограничивающих функций для мутировавшей клетки

$$\hat{s}_i = \sum_{z=1}^{Z_1} |h_z(\hat{x}_i)| + \sum_{z=1}^{Z_2} \max\{0, g_z(\hat{x}_i)\}.$$

6.6. Замена клетки мутировавшей клеткой

Если  $\hat{s}_{i1} < s_i$  или  $\hat{s}_{i1} = s_i \wedge F(\hat{x}_i) < F(x_i)$ , то

$$b_i = \hat{b}_i, x_i = \hat{x}_i, s_i = \hat{s}_i.$$

6.7. Если  $i < N_E$ , то  $i = i + 1$ , переход на шаг 6.3.

6.8. Если  $i < L_E$ , то  $i = i + 1$ , переход на шаг 6.2.

7. Вычисление значения адаптивного порога  $\Delta_E$  для  $P^E$

$$\Delta_E = \frac{1}{L_E} \sum_{k=1}^{L_E} s_k.$$

Если  $\Delta_E < 0,0001$ , то  $\Delta_E = 0,001$ .

8. Создание субпопуляции клеток памяти  $P^M$ .

8.1. Разбиение субпопуляции исполнительных клеток  $P^E$  на подмножество  $P1^E = \{(b1_k, x1_k, s1_k)\}$ , содержащее клетки, для которых  $s1_k < \Delta_E$ , и подмножество  $P2^E = \{(b2_k, x2_k, s2_k)\}$ , содержащее клетки, для которых  $s2_k \geq \Delta_E$ .

8.2. Упорядочение подмножества  $P1^E$  по функции цели, т.е.  $F(x1_k) < F(x1_{k+1})$ .

8.3. Упорядочение множества  $P2^E$  по сумме значений всех ограничивающих функций, т.е.  $s2_k < s2_{k+1}$ .

8.4. Если  $n = 1$ , то  $L_M$  первых клеток из упорядоченного объединения  $P1^E \cup P2^E$  образует субпопуляцию исполнительных клеток  $P^M = \{(x_i, s_i)\}$ , причем бинарное представление  $b_i$  для клеток памяти не используется.

Если  $n > 1$ , то  $L_M / 2$  первых клеток из упорядоченного объединения  $P1^E \cup P2^E$  заменяют  $L_M / 2$  худших (последних) клеток субпопуляции исполнительных клеток  $P^M$ , причем бинарное представление  $b_i$  для клеток памяти не используется.

9. Модификация субпопуляции клеток памяти  $P^M$  на основе мутации.

9.1. Номер клетки  $i = 1$ .

9.2. Номер мутации  $l = 1$ .

9.3. Выполнить мутацию

$$\hat{x}_{ij} = x_{ij} \pm \Delta = x_{ij} + \Delta(-1 + 2\text{round}(\text{rand}())),$$

$j \in \overline{1, M}$ ,

Шаг мутации  $\Delta$  вычисляется в виде

$$\Delta = \left( \frac{(x_j^{\max} - x_j^{\min})\text{rand}()}{10^\alpha n(Z_1 + Z_2)M} \right)^{\text{rand}()}.$$

9.4. Суммируются значения всех ограничивающих функций для мутировавшей клетки

$$\hat{s}_i = \sum_{z=1}^{Z_1} |h_z(\hat{x}_i)| + \sum_{z=1}^{Z_2} \max\{0, g_z(\hat{x}_i)\}.$$

9.5. Замена клетки мутировавшей клеткой

Если  $\hat{s}_{i1} < s_i$  или  $\hat{s}_{i1} = s_i \wedge F(\hat{x}_i) < F(x_i)$ , то  $x_i = \hat{x}_i$ ,  $s_i = \hat{s}_i$ .

9.6. Если  $l < N_M$ , то  $l = l + 1$ , переход на шаг 9.3.

9.7. Если  $i < L_M$ , то  $i = i + 1$ , переход на шаг 9.2.

10. Задание фиксированного порога  $\Delta_M$  для субпопуляции  $P^M$

$$\Delta_M = 0,0001.$$

11. Упорядочение субпопуляции клеток памяти  $P^M$ .

11.1. Разбиение субпопуляции клеток памяти  $P^M$  на подмножество  $P1^M = \{(x1_k, s1_k)\}$ , содержащее клетки, для которых  $s1_k < \Delta_M$ , и подмножество  $P2^M = \{(x2_k, s2_k)\}$ , содержащее клетки, для которых  $s2_k \geq \Delta_M$ .

11.2. Упорядочение подмножества  $P1^M$  по функции цели, т.е.  $F(x1_k) < F(x1_{k+1})$ .

11.3. Упорядочение множества  $P2^M$  по сумме значений всех ограничивающих функций, т.е.  $s2_k < s2_{k+1}$ .

$$P^M = P1^M \cup P2^M = \{(x_i, s_i)\}.$$

12. Если  $F(x_1) < F(x^*)$ , то  $x^* = x_1$ .

13. Если  $n < N$ , то  $n = n + 1$ , переход на шаг 3.

Результатом является  $x^*$ .

### 3. Алгоритм новой модели Т-клеток

Алгоритм новой модели Т-клеток, предложенный в данной статье и использующий только вещественное представление, состоит из следующих шагов:

1. Инициализация.

1.1. Задание параметра для мутации исполнительской клетки  $\beta$ , параметра для мутации клетки памяти  $\alpha$ , причем  $\alpha, \beta$  – натуральные числа.

1.2. Задание максимального числа итераций  $N$ , длины клетки  $M$ , размера субпопуляции новых клеток  $L_V$ , количества отбираемых новых клеток с учетом ограничений  $L1_V$ , количества отбираемых новых клеток без учета ограничений  $L2_V$ , количества мутаций каждой исполнительской клетки  $N_E$ , размера субпопуляции клеток памяти  $L_M$ , количества мутаций каждой клетки памяти  $N_M$ , причем  $L_V < L_M$ .

1.3. Задание минимальных и максимальных значений для клетки  $x_j^{\min}, x_j^{\max}$ ,  $j \in \overline{1, M}$ .

1.4. Задание функции стоимости (функция цели) и ограничений

$$F(x) \rightarrow \min_x,$$

$$h_z(x) = 0, z \in \overline{1, Z_1},$$

$$g_z(x) \leq 0, z \in \overline{1, Z_2},$$

где  $x$  – клетка (вещественный вектор),

$Z_1$  – количество ограничений-равенств,

$Z_2$  – количество ограничений-неравенств.

1.4. Создание случайным образом лучшей клетки

$$x^* = (x_1^*, \dots, x_M^*),$$

$$x_j^* = x_j^{\min} + (x_j^{\max} - x_j^{\min})\text{rand}(),$$

где  $\text{rand}()$  – функция, возвращающая равномерно распределенное случайное число в диапазоне  $[0, 1]$ .

2. Номер итерации  $n = 1$ .

3. Создание субпопуляции новых клеток  $P^V$ .

3.1. Номер клетки  $k = 1$ ,  $P^V = \emptyset$ .

3.2. Создание случайным образом клетки

$$x_k = (x_{k1}, \dots, x_{kM}),$$

$$x_{kj} = x_j^{\min} + (x_j^{\max} - x_j^{\min}) \text{rand}().$$

3.3. Суммируются значения всех ограничивающих функций для  $i$ -й клетки

$$s_k = \sum_{z=1}^{Z_1} |h_z(x_k)| + \sum_{z=1}^{Z_2} \max\{0, g_z(x_k)\}.$$

3.4. Если  $(x_k, s_k) \notin P^V$ , то

$$P^V = P^V \cup \{(x_k, s_k)\}, k = k + 1.$$

3.5. Если  $k \leq L_V$ , то переход на шаг 3.2.

4. Вычисление значения адаптивного порога  $\Delta_V$  для субпопуляции  $P^V$

$$\Delta_V = \frac{1}{L_V} \sum_{k=1}^{L_V} s_k.$$

Если  $\Delta_V < 0,0001$ , то  $\Delta_V = 0,1$ .

5. Создание субпопуляции исполнительных клеток  $P^E$ .

5.1. Разбиение субпопуляции новых клеток  $P^V$  на подмножество  $P1^V = \{(x1_k, s1_k)\}$ , содержащее клетки, для которых  $s1_k < \Delta_V$ , и подмножество  $P2^V = \{(x2_k, s2_k)\}$ , содержащее клетки, для которых  $s2_k \geq \Delta_V$ .

5.2. Упорядочение подмножества  $P1^V$  по функции цели, т.е.  $F(x1_k) < F(x1_{k+1})$ .

5.3. Упорядочение множества  $P2^V$  по сумме значений всех ограничивающих функций, т.е.  $s2_k < s2_{k+1}$ .

5.4.  $L1_V$  первых клеток из упорядоченного множества  $P1^V$  и  $L2_V$  первых клеток из упорядоченного множества  $P2^V$  образует субпопуляцию исполнительных клеток  $P^E = \{(x_i, s_i)\}$  мощностью  $L_E = L1_V + L2_V$ , причем первыми идут клетки из множества  $P1^V$ .

6. Модификация субпопуляции исполнительных клеток  $P^E$  на основе мутации.

6.1. Номер клетки  $i = 1$ .

6.2. Номер мутации  $l = 1$ .

6.3. Вычисление максимального значения всех ограничивающих функций

$$s^{\max} = \max\{|h_1(x_i)|, \dots, |h_{Z_1}(x_i)|, \max\{0, g_1(x_i)\}, \dots, \max\{0, g_{Z_2}(x_i)\}\}.$$

6.4. Выполнить мутацию

$$\text{Если } s^{\max} \leq \frac{s_i}{Z_1 + Z_2}, \text{ то}$$

$$\hat{x}_{ij} = x_{ij} \pm \Delta = x_{ij} + \Delta(-1 + 2\text{round}(\text{rand}())), j \in \overline{1, M},$$

где  $\text{round}()$  – функция, округляющая число до ближайшего целого.

Если  $s^{\max} > \frac{s_i}{Z_1 + Z_2}$ , то

$$\hat{x}_i = x_i,$$

$$j = \text{round}(1 + (M - 1)\text{rand}()),$$

$$\hat{x}_{ij} = x_{ij} \pm \Delta = x_{ij} + \Delta(-1 + 2\text{round}(\text{rand}())).$$

Шаг мутации  $\Delta$  вычисляется в виде

$$\lambda = \text{rand}(),$$

$$r = \text{rand}(),$$

$$\Delta = \begin{cases} (x_j^{\max} - x_{ij})r \left(1 - \frac{n}{N}\right)^\beta, & \lambda < 0,5; \\ -(x_{ij} - x_j^{\min})r \left(1 - \frac{n}{N}\right)^\beta, & \lambda \geq 0,5. \end{cases}$$

6.5. Суммируются значения всех ограничивающих функций для мутировавшей клетки

$$\hat{s}_i = \sum_{z=1}^{Z_1} |h_z(\hat{x}_i)| + \sum_{z=1}^{Z_2} \max\{0, g_z(\hat{x}_i)\}.$$

6.6. Замена клетки мутировавшей клеткой

Если  $\hat{s}_{i1} < s_i$  или  $\hat{s}_{i1} = s_i \wedge F(\hat{x}_i) < F(x_i)$ , то  $x_i = \hat{x}_i$ ,  $s_i = \hat{s}_i$ .

6.7. Если  $i < N_E$ , то  $i = i + 1$ , переход на шаг 6.3.

6.8. Если  $i < L_E$ , то  $i = i + 1$ , переход на шаг 6.2.

7. Вычисление значения адаптивного порога  $\Delta_E$  для  $P^E$

$$\Delta_E = \frac{1}{L_E} \sum_{k=1}^{L_E} s_k.$$

Если  $\Delta_E < 0,0001$ , то  $\Delta_E = 0,001$ .

8. Создание субпопуляции клеток памяти  $P^M$ .

8.1. Разбиение субпопуляции исполнительных клеток  $P^E$  на подмножество  $P1^E = \{(x1_k, s1_k)\}$ , содержащее клетки, для которых  $s1_k < \Delta_E$ , и подмножество  $P2^E = \{(x2_k, s2_k)\}$ , содержащее клетки, для которых  $s2_k \geq \Delta_E$ .

8.2. Упорядочение подмножества  $P1^E$  по функции цели, т.е.  $F(x1_k) < F(x1_{k+1})$ .

8.3. Упорядочение множества  $P2^E$  по сумме значений всех ограничивающих функций, т.е.  $s2_k < s2_{k+1}$ .

8.4. Если  $n = 1$ , то  $L_M$  первых клеток из упорядоченного объединения  $P1^E \cup P2^E$  образует субпопуляцию исполнительных клеток  $P^M = \{(x_i, s_i)\}$ .

Если  $n > 1$ , то  $L_M / 2$  первых клеток из упорядоченного объединения  $P1^E \cup P2^E$  заменяют  $L_M / 2$  худших (последних) клеток субпопуляцию исполнительных клеток  $P^M$ .

9. Модификация субпопуляции клеток памяти  $P^M$  на основе мутации.

9.1. Номер клетки  $i = 1$ .

9.2. Номер мутации  $l = 1$ .

9.3. Выполнить мутацию

$$\hat{x}_{ij} = x_{ij} \pm \Delta = x_{ij} + \Delta(-1 + 2\text{round}(\text{rand}())),$$

$j \in \overline{1, M}$ .

Шаг мутации  $\Delta$  вычисляется в виде

$$\Delta = \left( \frac{(x_j^{\max} - x_j^{\min}) \text{rand}()}{10^\alpha n(Z_1 + Z_2)M} \right)^{\text{rand}}.$$

9.4. Суммируются значения всех ограничивающих функций для мутировавшей клетки

$$\hat{s}_i = \sum_{z=1}^{Z_1} |h_z(\hat{x}_i)| + \sum_{z=1}^{Z_2} \max\{0, g_z(\hat{x}_i)\}.$$

9.5. Замена клетки мутировавшей клеткой.

Если  $\hat{s}_{i1} < s_i$  или  $\hat{s}_{i1} = s_i \wedge F(\hat{x}_i) < F(x_i)$ , то  $x_i = \hat{x}_i$ ,  $s_i = \hat{s}_i$ .

9.6. Если  $l < N_M$ , то  $l = l + 1$ , переход на шаг 9.3.

9.7. Если  $i < L_M$ , то  $i = i + 1$ , переход на шаг 9.2.

10. Задание фиксированного порога  $\Delta_M$  для субпопуляции  $P^M$ .

$$\Delta_M = 0,0001.$$

11. Упорядочение субпопуляции клеток памяти  $P^M$ .

11.1. Разбиение субпопуляции клеток памяти  $P^M$  на подмножество  $P1^M = \{(x1_k, s1_k)\}$ , содержащее клетки, для которых  $s1_k < \Delta_M$ , и подмножество  $P2^M = \{(x2_k, s2_k)\}$ , содержащее клетки, для которых  $s2_k \geq \Delta_M$ .

11.2. Упорядочение подмножества  $P1^M$  по функции цели, т.е.  $F(x1_k) < F(x1_{k+1})$ .

11.3. Упорядочение множества  $P2^M$  по сумме значений всех ограничивающих функций, т.е.  $s2_k < s2_{k+1}$ .

$$11.4. P^M = P1^M \cup P2^M = \{(x_i, s_i)\}.$$

12. Если  $F(x_1) < F(x^*)$ , то  $x^* = x_1$ .

13. Если  $n < N$ , то  $n = n + 1$ , переход на шаг 3.

Результатом является  $x^*$ .

#### 4. Сравнительный анализ метаэвристических методов

В табл. 1 приведены особенности предложенной новой модели Т-клеток и существующих метаэвристических методов, решающих задачу условной оптимизации числовых функций. Предполагается, что приведенный в табл. 1 генетический алгоритм

использует штрафную функцию. Как следует из табл. 1, предложенная новая модель Т-клеток использует вещественное представление клеток, шаг мутации зависит от номера итерации, а настройка параметров штрафной функции не требуется, что позволяет уменьшить вычислительную сложность метода и сделать поиск направленным.

Таблица 1

Особенности метаэвристических методов

Свойства методов	Генетический алгоритм для условной оптимизации	Классическая модель Т-клеток	Новая модель Т-клеток
Количество популяций	обычно 1	3	3
Представление особей популяции	зависит от конкретного алгоритма	бинарное и вещественное	только вещественное
Отбор в популяцию на основе функции цели и ограничений	+	+	+
Кроссовер присутствует	+	-	-
Количество типов мутации	1	2	2
Шаг мутации зависит от номера итерации	только в случае неоднородной мутации	только для второго типа мутации	для обоих типов мутации
Выполнения мутации	только над хромосомами, выбранными с определенной вероятностью	для всех клеток второй и третьей субпопуляции	для всех клеток второй и третьей субпопуляции
Требуется настройка параметров штрафной функции	+	-	-

#### Заключение

*Новизна.* В статье предложена новая модель Т-клеток, базирующаяся на механизме Т-клеток иммунной системы и неоднородной мутации генетического алгоритма. В отличие от классической модели Т-клеток, данная метаэвристика использует только вещественное представление клеток, что уменьшает вычислительную сложность алгоритма, а для субпопуляции исполнительных клеток использует неоднородную мутацию, шаг которой уменьшается на каждой итерации, что позволяет на заключительных стадиях работы алгоритма сделать поиск направленным.

*Практическое значение.* Основные положения данной работы предназначены для реализации в интеллектуальных системах управления и прогноза

для поиска оптимального управляющего воздействия и идентификации параметров.

### Литература

1. Федоров, Е. Е. Усовершенствование метаэвристических методов для решения задачи поиска оптимального маршрута [Текст] / Е. Е. Федоров, О. П. Мартынова, Э. Слесорайтите // Наукові праці Донецького національного технічного університету. – 2013. – № 1. – С. 117–124.

2. Brownlee, J. *Clever Algorithms: Nature-Inspired Programming Recipes* [Text] / J. Brownlee. – Melbourne : Brownlee, 2011. – 436 p.

3. *Evolutionary Computation 2. Advanced Algorithms and Operators* [Text] / ed. by T. Back, D. B. Fogel, Z. Michalewicz. – Bristol : IOP Publishing Ltd., 2000. – 304 p.

4. Cortes, N. C. *Handling constraints in global optimization using artificial immune system* [Text] / N. C. Cortes, D. Trejo-Perez, C. A. Coello Coello // Proc. 4th International Conference ICARIS 2005, 10-12 August 2005. – Banff, 2005. – P. 234–247.

5. Chueh, H. *MOIA: Multi Objective Immune Algorithm* [Text] / H. Chueh, G. C. Luh, W. W. Liu // *Engineering Optimization*. – 2003. – Vol. 35, № 2. – P. 143–164.

6. Aragon, V. S. *Solving Constrained Optimization using a T-Cell Artificial Immune System* [Text] / V. S. Aragon, S. C. Esquivel, C. A. Coello Coello // *Inteligencia Artificial*. – 2008. – Vol. 40. – P. 7–22.

7. Yoo, J. *Enhanced GA Based Search Through Immune System Modeling* [Text] / J. Yoo, P. Hajela // Proc. 3rd World Congress on Structural and Multidisciplinary Optimization, 20-21 May 1999. – New York, 1999. – P. 310-315.

8. Coello Coello, C.A. *Hybridizing a genetic algorithm with an artificial immune system for global optimization* [Text] / C. A. Coello Coello, N. C. Cortes // *Engineering Optimization*. – 2004. – Vol. 36, № 5. – P. 607–634.

9. Bernardino, H. S. *Constraints handling in genetic algorithms via artificial immune systems* [Text] / H. S. Bernardino, H. J. C. Barbosa, A. C. C. Lemonge // Proc. Genetic and Evolutionary Computation Conference (GECCO 2006), 25-26 July 2006. – Seattle, 2006. – P. 646-653.

Поступила в редакцію 10.04.2013, рассмотрена на редколлегии 19.05.2014

**Рецензент:** д-р техн. наук, проф., проф. каф. электронной техники Н. И. Чичикало, ГБУЗ «Донецкий Национальный технический университет», Донецк.

### РОЗРОБКА НОВОЇ МОДЕЛІ Т-КЛІТИН ДЛЯ ВИРІШЕННЯ ЗАДАЧ УМОВНОЇ ОПТИМІЗАЦІЇ

*Є. Є. Федоров*

Для створення інтелектуальної системи керування й прогнозу в статті було запропоновано метаевристичний метод, заснований на принципах моделі Т-клітин імунних систем і неоднорідної мутації, що використовується в генетичному алгоритмі. Запропонована нова модель Т-клітин дозволяє на ранніх стадіях роботи алгоритму досліджувати весь простір пошуку, а на заключних стадіях зробити пошук спрямованим. Крім того, ця модель дозволяє використати тільки речовинне подання кліток (потенційних рішень), що дозволяє зменшити обчислювальну складність. Розроблений метод було зрівняно з аналогічними методами умовної оптимізації числових функцій.

**Ключові слова:** інтелектуальна система керування й прогнозу, нова модель Т-клітин, імунна система, генетичний алгоритм, умовна оптимізація числових функцій, неоднорідна мутація, метаевристика.

### DEVELOPMENT OF NEW T-CELL MODEL FOR THE DECISION OF PROBLEMS OF CONSTRAINED OPTIMIZATION

*E. E. Fedorov*

For creation of an intellectual control system and the forecast in article the metaheuristic method based on principles of T-cell model of immune systems and a non-uniform mutation which is used in genetic algorithm has been offered. The offered new T-cell model allows to investigate all space of search at early stages of work of algorithm, and at final stages to make search directed. Besides, this model allows to use only real representation of cell (potential decisions) that allows to reduce computing complexity. The developed method has been compared to similar methods of constrained optimization of numerical functions.

**Key words:** an intellectual control system and the forecast, new T-cell model, immune system, genetic algorithm, constrained optimization of numerical functions, a non-uniform mutation, metaheuristics.

**Федоров Евгений Евгеньевич** – д-р техн. наук, доцент, заведуючий кафедрой специализированных компьютерных систем, Донецкая академия автомобильного транспорта, Донецк, Украина, e-mail: fee75@mail.ru.