

**В. П. СЕВЕРИН**, д-р техн. наук, проф. НТУ «ХПИ»;  
**Е. Н. НИКУЛИНА**, канд. техн. наук, ст. преп. НТУ «ХПИ»;  
**Т. Е. НИКОЛАЕНКО**, студентка НТУ «ХПИ»

## МОДИФИКАЦИЯ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ ОПТИМИЗАЦИИ ВЕКТОРНЫХ ЦЕЛЕВЫХ ФУНКЦИЙ

Запропоновані модифікації генетичних алгоритмів для оптимізації векторних цільових функцій. Приведені модифіковані бінарний та безперервний генетичні алгоритми. В модифікованих генетичних алгоритмах використовуються значення векторних функцій.

**Ключові слова:** генетичні алгоритми, векторні цільові функції, оптимізація.

Предложены модификации генетических алгоритмов для оптимизации векторных целевых функций. Приведены модифицированные бинарный и непрерывный генетические алгоритмы. В модифицированных генетических алгоритмах используются значения векторных функций.

**Ключевые слова:** генетические алгоритмы, векторные целевые функции, оптимизация.

Modification of the genetic algorithms are proposed for optimization of vector objective functions. Modified binary and continuous genetic algorithms are given. Values of the vector functions are used in modified genetic algorithms.

**Keywords:** genetic algorithms, vector objective functions, optimization.

**Введение.** Для многокритериального синтеза систем автоматического управления (САУ) путем оптимизации векторных целевых функций модифицируются методы оптимизации [1, 2]. Проекция векторных целевых функций не являются непрерывными функциями. К тому же задачи оптимизации параметров интеллектуальных регуляторов, как правило, являются многоэкстремальными задачами оптимизации. Для повышения надежности предлагаемых методов синтеза оптимизацию векторных целевых функций целесообразно производить генетическими алгоритмами (ГА) [3, 4].

Цель статьи заключается в модификации генетических алгоритмов для оптимизации векторных целевых функций. Для достижения поставленной цели рассматриваются основные принципы ГА и выполняется их модификация с учетом особенностей задач синтеза САУ.

**Основы генетических алгоритмов.** Каждую особь популяции представим хромосомой  $\xi = (\xi_1, \xi_2, \dots, \xi_p)$ , состоящей из бинарных генов:

$\xi_i = (\xi_{i1}, \xi_{i2}, \dots, \xi_{im})$ ,  $i = \overline{1, p}$ ,  $\xi_{ij} \in \{0, 1\}$ . Здесь  $m$  – количество битов в гене, то есть длина двоичной последовательности гена. Хромосома представляется бинарной строкой длины  $M = mp$ . Хромосоме  $\xi$ , определяющей генотип особи, отвечает вектор переменных параметров  $x = (x_1, x_2, \dots, x_p) \in R^p$ ,

соответствующий фенотипу особи. Пусть  $x_i \in [a_i, b_i] \in R$ ,  $i = \overline{1, p}$ , и требуется найти решение с точностью до  $q$  знаков после запятой для каждой переменной  $x_i$ . Для этого необходимо интервал  $[a_i, b_i]$  длины  $l_i = b_i - a_i$  разбить на  $10^q l_i$  одинаковых подынтервалов с шагом дискретизации  $r = 10^{-q}$ . Длину двоичной последовательности, требуемой для кодирования десятичного числа  $x_i$  из интервала  $[a_i, b_i]$  с шагом  $r$ , определяет наименьшее натуральное число  $m$ , удовлетворяющее системе неравенств  $10^q l_i \leq 2^m - 1$ ,  $i = \overline{1, p}$ . Решая эту систему неравенств, получим длину гена

$$m = \max_i [\log_2(10^q l_i + 1)] + 1. \quad (1)$$

В таблице представлены значения рассчитанной по формуле (1) длины гена  $m$  для различных значений параметра точности  $q$  и одинаковых длин интервалов  $l = l_i$  переменных параметров.

Зависимость длины гена от параметра точности и длины интервала

$q$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$l = 1$	4	7	10	14	17	20	24	27	30	34	37	40	44	47
$l = 10$	7	10	14	17	20	24	27	30	34	37	40	44	47	50
$l = 100$	10	14	17	20	24	27	30	34	37	40	44	47	50	54

Для получения решения с восьмью знаками после запятой с  $q = 8$  необходимо использовать при длине интервала  $l = 1$  длину гена  $m = 27$ , а при  $l = 100$  –  $m = 34$ . Положим в вычислительных экспериментах  $m = 32$ .

Значению гена  $\xi_i$  соответствует десятичное представление числа

$$d_i = \xi_{i1} 2^{m-1} + \xi_{i2} 2^{m-2} + \dots + \xi_{im} 2^0. \quad (2)$$

Значение переменного параметра представляется равенством

$$x_i = a_i + l_i d_i / (2^m - 1). \quad (3)$$

Таким способом задаются фенотипы, соответствующие кодовым последовательностям с длиной  $m$ . Выражения (2) и (3) – это следствие линейного отображения интервала  $[0, 2^m - 1]$  на интервал  $[a_i, b_i]$ . Иногда вместо двоично-десятичного кода применяется код Грея.

Популяция особей является массивом хромосом  $\Pi = (\xi^{(1)}, \xi^{(2)}, \dots, \xi^{(N)})$ , где  $N$  – число особей, определяющее размер популяции. Популяция

хромосом отображается по формулам (2) и (3) в популяцию точек  $P = (x^{(1)}, x^{(2)}, \dots, x^{(N)})$  пространства переменных параметров  $R^p$ .

**Модификация генетических алгоритмов.** В отличие от обычных ГА, использующих в качестве функции приспособленности скалярную целевую функцию, в модифицированных ГА используется массив значений векторной целевой функции  $\Phi = (F^{(1)}, F^{(2)}, \dots, F^{(N)})$ . Таким образом, каждая особь с номером  $j \in \overline{1, N}$  представляется хромосомой  $\Pi_j = \xi^{(j)}$ , точкой  $P_j = x^{(j)}$  пространства параметров и значением векторной целевой функции  $\Phi_j = F^{(j)}$ .

Генетические алгоритмы являются итерационными, и каждая итерация в них называется эпохой. На  $k$ -той эпохе формируется популяция особей, которой соответствуют массивы хромосом  $\Pi^{(k)}$ , точек  $P^{(k)}$  и значений целевой функции  $\Phi^{(k)}$ . Массив хромосом начальной популяции  $\Pi^{(0)}$  формируется случайным образом. Последующие популяции формируются с применением генетических операторов: селекции, кроссовера и мутации.

Для выбора родительских особей применяется комбинация ранговой селекции с принципом рулетки [3, 4]. Ранговая селекция основана на ранжировании особей по значениям целевой функции. При этом списки особей  $\Pi$ ,  $P$  и  $\Phi$  сортируются с использованием операций сравнения значений векторных функций [1, 2], то есть упорядочиваются от более приспособленных особей с лучшими значениями целевой функции к менее приспособленным особям с худшими значениями функции. Из более приспособленных особей выбираются  $N_p$  хромосом родительских особей  $\Pi_p$ ,  $N_p < N$ . Каждой родительской особи по ее месту  $j$  в списке приписывается ранг  $r_j = N_p + 1 - j$ ,  $j \in \overline{1, N_p}$ . Сумма всех рангов равна сумме арифметической прогрессии  $S_r = 0,5(N_p + 1)N_p$ . Колесу рулетки сопоставляется интервал  $[0; 1]$ , а каждой родительской особи – сектор длиной  $p_j = r_j/S_r$ , соответствующий вероятности попадания особи в сектор. По принципу рулетки из  $N_p$  родительских особей выбирается  $n_p = [0,5 \cdot (N - N_p)]$  номеров первых  $N_1$  и вторых  $N_2$  родителей для скрещивания. По этим номерам одноточечным кроссовером над массивом родительских особей  $\Pi_p$  формируются массивы хромосом потомков  $\pi_1$  и  $\pi_2$ . Формируется промежуточная популяция хромосом как объединение массивов  $\Pi_p$ ,  $\pi_1$  и  $\pi_2$ , которая подвергается мутации и выполняется переход к новой популяции. Бинарный ГА представим следующим алгоритмом.

**Модифицированный бинарный ГА.** Входные параметры:  $a$  и  $b$  – векторы ограничений,  $N$  – размер популяции,  $m$  – длина гена,  $r_p$  – доля родительских пар в популяции,  $r_m$  – доля мутации,  $N_k$  – предельное число итераций. Выходные параметры:  $x$  и  $F_x$  – лучшая точка и значение в ней векторной целевой функции. **1.** Положить  $k=1$ ,  $p = \dim(a)$ ,  $l = b - a$ ,  $M = m \cdot p$ . **2.** Вычислить  $N_p = [r_p \cdot N]$ ,  $n_p = [0,5 \cdot (N - N_p)]$ ,  $n_m = [(N - 1) \cdot r_m \cdot M]$ . **3.** Вычислить  $\Pi = \text{Round}(\text{Rand}(N, M))$ . **4.** Вычислить  $P = \text{Decod}(\Pi, a, l, m)$ . **5.** Для  $j \in \overline{1, N}$  вычислить  $\Phi_j = F(P_j)$ . **6.** Положить  $(\Phi, J) = \text{Sort}(\Phi)$ ,  $\Pi = \Pi(J)$ ,  $P = P(J)$ ,  $\Pi_p = \Pi(1 : N_p)$ . **7.** Вычислить  $(N_1, N_2) = \text{Roulette}(N_p, n_p)$ . **8.** Вычислить  $(\pi_1, \pi_2) = \text{Crossover}(\Pi_p, N_1, N_2)$ . **9.** Положить  $\Pi = (\Pi_p, \pi_1, \pi_2)$ ,  $\Pi = \text{Mutate}(\Pi, n_m)$ . **10.** Вычислить  $P = \text{Decod}(\Pi, a, l, m)$ . **11.** Для  $j \in \overline{1, N}$  вычислить  $\Phi_j = F(P_j)$ . **12.** Положить  $(\Phi, J) = \text{Sort}(\Phi)$ ,  $\Pi = \Pi(J)$ ,  $P = P(J)$ ,  $\Pi_p = \Pi(1 : N_p)$ . **13.** Положить  $x = P_1$ ,  $F_x = \Phi_1$ . **14.** Если  $k < N_k$ , положить  $k = k + 1$  и перейти к п. 7. **15.** Выход.

В этом алгоритме на шагах 1 и 2 инициализируются основные параметры ГА: равное размерности вектора  $a$  число переменных параметров  $p$ , вектор длин интервалов переменных параметров  $l$ , длина хромосомы  $M$ , число родительских особей  $N_p$ , число пар родителей  $n_p$ , количество мутаций  $n_m$ , счетчик числа итераций  $k$ . На шаге 3 создается начальная популяция хромосом  $\Pi$  путем формирования процедурой  $\text{Rand}$  матрицы размерности  $N \times M$  случайных вещественных чисел из интервала  $[0; 1]$  и округления этих чисел до ближайших целых значений 0 или 1 процедурой  $\text{Round}$ . На шаге 4 популяция хромосом декодируется по формулам (2) и (3) процедурой  $\text{Decod}$  в множество точек  $P$  пространства переменных параметров  $R^p$ . На шаге 5 вычисляются значения векторной целевой функции для всех особей популяции. На шаге 6 выполняется сортировка особей по значениям проекций вектора целевой функции  $\Phi$  в порядке ухудшения значений. Применяется специально разработанная для значений векторной целевой функции процедура пузырьковой сортировки  $\text{Sort}$ , основанная на операциях сравнения [1, 2]. В результате выполнения этой процедуры получен отсортированный вектор значений функции  $\Phi$  и соответствующий ему массив исходных индексов проекций  $J$ , который используется для сортировки массивов хромосом  $\Pi$  и соответствующих им точек  $P$ . Из массива хромосом  $\Pi$  выбирается  $N_p$  первых хромосом, составляющие «родительский пул»  $\Pi_p$ . На шаге 7 по принципу рулетки с

ранговой селекцией процедурой *Roulette* из  $N_p$  родительских особей выбирается  $n_p$  номеров первых  $N_1$  и вторых  $N_2$  родителей для скрещивания. С хромосомами выбранных родителей процедурой *Crossover* выполняется одноточечный кроссовер и формируются массивы хромосом потомков  $\pi_1$  и  $\pi_2$ . На шаге 8 из родительских особей и потомков формируется промежуточная популяция хромосом. На шаге 9 выполняется мутация этой популяции процедурой *Mutate*. На шагах 10–12 опять выполняется декодирование хромосом, вычисление значений целевой функции и сортировка популяции. На шаге 13 определяется лучшая точка текущей популяции. На шаге 14 проверяется критерий завершения ГА по достижению счетчиком итераций  $k$  предельного количества итераций  $N_k$ . Итерацию ГА составляют шаги 7–14. По умолчанию задаются значения входных параметров алгоритма  $r_p = 0,5$ ,  $r_m = 0,15$ .

Наряду с бинарными ГА применяются непрерывные ГА, в которых исходной является не популяция двоичных хромосом  $\Pi$ , а популяция точек  $P$  пространства переменных параметров задачи  $R^p$ . В этом случае ГА упрощается, поскольку исчезает необходимость декодирования хромосом. Массив точек начальной популяции  $P^{(0)}$  формируется случайным образом. Последующие популяции формируются с применением генетических операторов: ранговой селекции с принципом рулетки, а также непрерывных кроссовера и мутации [4]. Непрерывный ГА представим алгоритмом.

**Модифицированный непрерывный ГА.** Входные параметры:  $a$  и  $b$  – векторы нижних и верхних ограничений переменных,  $N$  – размер популяции,  $r_p$  – доля родительских пар,  $r_m$  – доля мутации,  $N_k$  – предельное количество итераций. Выходные параметры:  $x$  и  $F_x$  – лучшая точка и значение в ней векторной целевой функции. **1.** Положить  $p = \dim(a)$ ,  $l = b - a$ ,  $N_p = [r_p \cdot N]$ ,  $n_p = [0,5 \cdot (N - N_p)]$ ,  $n_m = [(N - 1) \cdot r_m \cdot p]$ ,  $k = 1$ . **2.** Для  $j = \overline{1, N}$  вычислить  $P_j = a + l \cdot \text{Rand}(1, p)$ ,  $\Phi_j = F(P_j)$ . **3.** Положить  $(\Phi, J) = \text{Sort}(\Phi)$ ,  $P = P(J)$ ,  $P_p = P(1 : N_p)$ . **4.** Вычислить  $(N_1, N_2) = \text{Roulette}(N_p, n_p)$ .  $(p_1, p_2) = \text{CrossoverC}(P_p, N_1, N_2)$ . **5.** Положить  $P = (P_p, p_1, p_2)$ . **6.** Вычислить  $P = \text{MutateC}(P, n_m)$ . **7.** Для  $j = \overline{1, N}$  вычислить  $\Phi_j = F(P_j)$ . **8.** Положить  $(\Phi, J) = \text{Sort}(\Phi)$ ,  $P = P(J)$ ,  $P_p = P(1 : N_p)$ . **9.** Положить  $x = P_1$ ,  $F_x = \Phi_1$ . **10.** Если  $k < N_k$ , положить  $k = k + 1$  и перейти к п. 4. **11.** Выход.

Здесь на шаге 1 инициализируются основные параметры ГА: число переменных параметров  $p$ , вектор длин интервалов переменных параметров

$l$ , число родительских особей  $N_p$ , число пар родителей  $n_p$ , количество мутаций  $n_m$ , счетчик числа итераций  $k$ . На шаге 2 случайным образом создается начальная популяция точек  $P$  с использованием процедуры *Rand* и вычисляется массив значений векторной целевой функции  $\Phi$ . На шаге 3 выполняется сортировка особей по массиву  $\Phi$  процедурой *Sort*, и по отсортированному множеству индексов  $J$  сортируется массив точек  $P$ . Из массива  $P$  выбирается  $N_p$  первых точек, составляющие «родительский пул»  $P_p$ . На шаге 4 процедурой *Roulette* из  $N_p$  родительских особей выбирается  $n_p$  номеров первых  $N_1$  и вторых  $N_2$  родителей для скрещивания. С координатами точек выбранных родителей процедурой *CrossoverC* выполняется одноточечный кроссовер и формируются массивы точек потомков  $p_1$  и  $p_2$ . На шаге 5 из родительских особей и потомков формируется промежуточная популяция точек. На шаге 6 выполняется непрерывная мутация этой популяции процедурой *MutateC*. На шагах 7–8 вычисляются значения целевой функции, и популяция сортируется. На шаге 9 определяется лучшая точка популяции. На шаге 10 проверяется критерий завершения ГА. Итерационный цикл ГА составляют шаги 4–10. Значения входных параметров алгоритма задаются  $r_p = 0,5$ ,  $r_m = 0,2$ .

**Выводы.** Рассмотрены основы построения генетических алгоритмов. Предложены модификации генетических алгоритмов для оптимизации векторных целевых функций. Приведены модифицированные бинарный и непрерывный генетические алгоритмы. В модифицированных генетических алгоритмах используются значения векторных целевых функций.

**Список литературы:** **1.** *Северин В. П.* Методы анализа и синтеза систем автоматического управления энергоблоков атомных электростанций / *В. П. Северин* // Проблемы обеспечения безопасности информационных и управляющих систем АЭС. – Одесса : Астропринт. – 2010. – С. 137–152. **2.** *Северин В. П.* Синтез систем автоматического управления энергоблока АЭС с нечеткими регуляторами генетическими алгоритмами / *В. П. Северин, С. М. Х. Джафаров, В. Ф. Чернай* // Технічна електродинаміка. Тематичний випуск «Силова електроніка та енергоефективність». – Ч. 1. – 2010. – С. 147–152. **3.** *Панченко Т. В.* Генетические алгоритмы. Учебное пособие / *Т. В. Панченко*. – Астрахань : Изд. дом «Астраханский университет», 2007. – 88 с. **4.** *Alireza M.* Genetic algorithms / *M. Alireza*. – Tehran : Naghoos, 2008. – 144 p. **5.** *Severin V. P.* Application of Genetic Algorithms to Vector Optimization of the Automatic Control Systems / *V. P. Severin* // Intelligent Information and Engineering Systems. Information Science and Computing. Information Technologies and Knowledge. – Vol. 3. – 2009. – N. 13. – P. 90–97. **6.** *Джафаров С. М. Х.* Многокритериальный синтез нечетких систем автоматического управления генетическими алгоритмами / *С. М. Х. Джафаров, В. П. Северин, Е. Н. Никулина* // Технічна електродинаміка. Тематичний випуск «Силова електроніка та енергоефективність». – Ч. 2. – 2010. – С. 149–154.

Надійшла до редакції 10.12.2012