УДК 519.95

К. А. Пупков, В. А. Феоктистов

ИССЛЕДОВАНИЕ ПРОБЛЕМЫ НЕЛИНЕЙНОЙ ОПТИМИЗАЦИИ В ЗАДАЧАХ ТЕХНИЧЕСКОГО ПРОЕКТИРОВАНИЯ

Рассмотрены проблемы нелинейной оптимизации при техническом проектировании, когда переменные принимают целочисленные, дискретные и непрерывные значения. Проведено исследование и дана количественная оценка различных методов нелинейной оптимизации. Предложен оригинальный подход к обработке ограничений, при котором нет необходимости усложнять целевую функцию штрафами. Показана быстрая сходимость и робастность метода.

При применении нелинейного программирования обычно предполагается, что переменные целевой функции непрерывны. Однако в реальных условиях при техническом проектировании часто переменные принимают как дискретные, так и целочисленные значения. Обычно они принимают дискретные значения, что связано с введением стандартизации при проектировании. Например, толщина стальной пластины, диаметр медной трубки, размер шайбы, модуль зубчатой передачи и т.д. часто ограничены набором стандартных доступных размеров. Целые величины часто выражают количество используемых единиц при проектировании, например число зубьев зубчатой передачи, количество болтов или заклепок, необходимых для фиксации конструкции, количество приводных ремней, используемых для трансмиссии, количество витых пружин и т.д.

При техническом проектировании необходимо решать проблемы нелинейной оптимизации, когда переменные принимают целые–дискретные–непрерывные значения. В современной литературе много внимания уделяется проблемам непрерывной оптимизации, однако на практике часто переменные принимают как дискретные, так и целые значения. Как правило, в таких случаях решают задачу непрерывной оптимизации (релаксации переменных), а затем округляют полученные значения до ближайшего целого или дискретного с учетом ограничений. При таком подходе конечный результат часто бывает далек от оптимального. В настоящее время не существует удовлетворительного универсального метода, применимого и к целым, и к дискретным, и к непрерывным переменным, при этом эффективного, надежного и простого в использовании.

В течение последнего десятилетия было предложено много интересных методов решения проблемы нелинейной оптимизации со смешанными переменными. Некоторые из них приведены в табл. 1. Однако не существует единого подхода, полностью соответствующего всем аспектам смешанного нелинейного программирования. Подчеркнем, что большинство современных методов имеют по крайней мере один из следующих недостатков: сложность реализации и использования, отсутствие гибкости, высокие вычислительные затраты, низкую надежность, скудные возможности оптимизации с ограничениями, невозможность нахождения допустимого решения.

Таблица 1

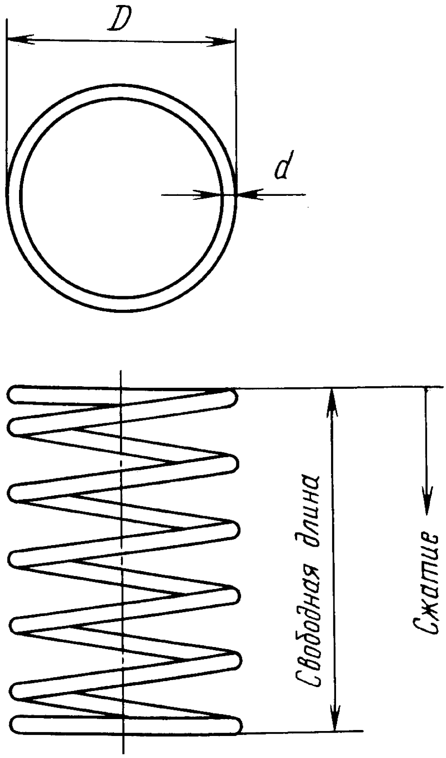
Автор(ы), предложивший(ие) метод	Методы и алгоритмы решения	Год
Сандгрен (Sandgren)	Метод "ветвей и границ" с использованием последовательного квадратичного программирования	1990
Фу, Фентон, Глегхорн (Fu, Fenton, Gleghorn)	Метод целого–дискретного–непрерывного нелинейного программирования	1991
Ло, Параламброс (Loh, Paralambros)	Алгоритм последовательной линеаризации	1991
Чанг, Ванг (Zhang, Wang)	Метод имитации отжига	1993
Чен, Тсао (Chen, Tsao)	Генетические алгоритмы	1993
Ли, Коу (Li, Choy)	Метод нелинейного смешанного дискретного программирования	1994
Ву, Коу (Wu, Choy)	Метагенетические алгоритмы	1995
Лин, Чанг, Ванг (Lin, Zhang, Wang)	Модифицированные генетические алгоритмы	1995
Тирауф, Саи (Thierauf, Cai)	Алгоритм двухуровневых параллельных эволюционных стратегий	1997
Као, Ву (Cao, Wu)	Алгоритм эволюционного программирования	1997
Лампинен, Зелинка (Lampinen, Zelinka)	Метод дифференциальной эволюции	1999

В работе [1] предложен оригинальный метод смешанной целой–дискретной–непрерывной нелинейной оптимизации для технического проектирования, основанный на алгоритме дифференциальной эволюции [2]. На примере конструирования витой пружины с исходными данными, взятыми из работы [3], покажем возможности этого метода. Кроме того, используя результаты работы [4], продемонстрируем оригинальный, естественный и простой алгоритм обработки нелинейных ограничений. В заключение приведем сравнительные характеристики рассмотренных методов.

Формулировка проблемы. В качестве проектируемого элемента выбрана спиральная пружина, работающая на сжатие (см. рисунок). Цель оптимизации — минимизировать объем стального провода, необходимый для изготовления одной пружины с заданными характеристиками.

К конструкционным параметрам пружины относятся: число N витков пружины; внешний диаметр D пружины; диаметр d стальной проволоки.

Число витков пружины — это переменная, принимающая целые значения. Внешний диаметр может быть любым, следовательно, используется непрерывная переменная. Диаметр стальной проволоки строго стандартизирован, и, следовательно, переменная d является дискретной.



Спиральная пружина, работающая на сжатие

Проблема согласно работе [3] формулируется следующим образом: необходимо найти значение

$$X = (x_1, x_2, x_3) = (N, D, d), \tag{1}$$

минимизировав выражение

$$f(X) = \frac{\pi^2 x_2 x_3^2 (x_1 + 2)}{4}$$

с учетом ограничений

$$g_1(X) = \frac{8C_f F_{\max} x_2}{\pi x_3^3} - S \leqslant 0, \quad g_2(X) = l_f - l_{\max} \leqslant 0,$$

$$g_3(X) = d_{\min} - x_3 \leqslant 0, \quad g_4(X) = x_2 - D_{\max} \leqslant 0,$$

$$g_5(X) = 3 - \frac{x_2}{x_3} \leqslant 0, \quad g_6(X) = \sigma_p - \sigma_{pm} \leqslant 0,$$

$$g_7(X) = \sigma_p + \frac{F_{\max} - F_p}{K} + 1{,}05\,(x_1 + 2)\,x_3 - l_f \leqslant 0,$$

$$g_8(X) = \sigma_p - \frac{F_{\max} - F_p}{K} \leqslant 0,$$

где $\sigma_p = F_p/K$ — текущая длина пружины,

$$C_f = \frac{4\left(\frac{x_2}{x_3}\right) - 1}{4\left(\frac{x_2}{x_3}\right) - 4} + \frac{0{,}615 x_3}{x_2} \leq 0,$$

$$K = \frac{G x_3^4}{8 x_1 x_2^3}, \quad l_f = \frac{F_{\max}}{K} + 1{,}05\,(x_1 + 2)\,x_3.$$

С помощью целевой функции $f(X)$ вычисляется объем пружины как функция конструкционных переменных. При этом учитываются следующие проектировочные параметры и ограничения: максимальная рабочая нагрузка $F_{\max} = 4448{,}222$ Н; максимально допустимое напряжение при сдвиге $S = 27{,}41213006$ Па; максимальная свободная длина $l_{\max} = 355{,}6$ мм; минимальный диаметр проволоки $d_{\min} = 5$ мм; максимальный внешний диаметр пружины $D_{\max} = 76{,}2$ мм; предварительная нагрузка сжатия $F_p = 1334{,}4666$ Н; максимально допустимое изменение длины под предварительной нагрузкой $\sigma_{pm} = 152{,}4$ мм; модуль сдвига материала $G = 11{,}5 \cdot 10^6$; общее отклонение должно быть согласовано с длиной пружины, т.е. при максимальной нагрузке соседние кольца не должны касаться друг друга; ограничения на продольный изгиб игнорируются; внешний диаметр D пружины должен быть по крайней мере в три раза больше, чем диаметр d стальной проволоки, чтобы избежать ущербных витков.

Метод дифференциальной эволюции. Впервые метод дифференциальной эволюции (ДЭ) был предложен в работе [2]. Метод ДЭ может быть классифицирован как эволюционный алгоритм оптимизации с плавающей точкой [1]. К настоящему времени разработано несколько стратегий ДЭ. В настоящей работе рассмотрим схему DE/rand/1/bin, подробное описание которой можно найти в работе [2]. Поскольку первоначально метод ДЭ был создан для решения задач, содержащих непрерывные переменные, то вначале рассмотрим непрерывную оптимизацию, а затем — обработку целых и дискретных переменных.

Пусть критерий оптимальности f имеет вид

$$f(X)\colon \mathbf{R}^n \to \mathbf{R}, \tag{2}$$

$$f(X) \to \min, \tag{3}$$

т.е. задача оптимизации — минимизировать целевую функцию $f(X)$ путем определения значений ее параметров

$$X = (x_1, \ldots, x_n) \in \mathbf{R}^n. \tag{4}$$

Как правило, параметры целевой функции ограничены своими предельными значениями L и H:

$$l_j \leqslant x_j \leqslant h_j, \quad l_j \in L, \quad h_j \in H, \quad j = 1, \ldots, n. \tag{5}$$

Как и все эволюционные алгоритмы оптимизации, метод ДЭ использует популяцию решений. Популяция P поколения G содержит N_P векторов решений — так называемых индивидуумов популяции. Каждый такой вектор представляет собой потенциальное решение проблемы оптимизации:

$$P^{(G)} = X_i^{(G)}, \quad i = 1, \ldots, N_P, \quad G = 1, \ldots, G_{\max}. \tag{6}$$

Каждый из N_P индивидуумов популяции P содержит n параметров (хромосом индивидуума):

$$P^{(G)} = X_i^{(G)} = x_{ij}^{(G)}, \quad i = 1, \ldots, N_P, \quad j = 1, \ldots, n. \tag{7}$$

Для инициализации популяции используется естественный способ случайного разброса индивидуумов при заданных начальных значениях

$$P^{(0)} = x_{ij}^{(0)} = \mathrm{rand}_{ij}\,(h_j - l_j) + l_j, \quad i = 1, \ldots, N_P, \quad j = 1, \ldots, n, \tag{8}$$

где rand — функция, генерирующая случайные значения, равномерно распределенные на интервале [0, 1].

Схема репродуцирования популяции в методе ДЭ отличается от схем в остальных эволюционных алгоритмах. Начиная с первого поколения $P^{(1)}$, каждое последующее поколение популяции $P^{(G+1)}$ воспроизводит себя на основании предыдущего $P^{(G)}$, но сначала формируется промежуточное поколение $P'^{(G+1)} = U_i^{(G+1)} = u_{ij}^{(G+1)}$:

$$u_{ij}^{(G+1)} = \begin{cases} x_{C_ij}^{(G)} + F\left(x_{A_ij}^{(G)} - x_{B_ij}^{(G)}\right), & \text{если} \quad (\mathrm{rand}_{ij} \leqslant C_r) \vee (j = D_i), \\ x_{ij}^{(G)} & \text{иначе;} \end{cases} \tag{9}$$

здесь

$$\begin{aligned} &A_i, B_i, C_i = \mathrm{rand}_{ij}\,[1, N_P], \\ &A_i \neq B_i \neq C_i \neq i, \\ &D_i = \mathrm{rand}_{ij}\,[1, n], \\ &i = 1, \ldots, N_P, \;\; j = 1, \ldots, n, \\ &C_r \in [0, 1]\,, \;\; F \in [0, 2] \subset \mathbf{R}, \end{aligned}$$

где A_i, B_i, C_i — три случайно выбранных отличных друг от друга индекса, т.е. три случайно выбранных индивидуума популяции (они также отличны от индекса текущего индивидуума, подлежащего изменениям); D_i — индекс случайно выбранной хромосомы каждого индивидуума, т.е. индивидуум следующего поколения отличается хотя бы на одну хромосому от индивидуума предыдущего поколения; C_r и F — управляющие параметры, значения которых остаются неизменными в процессе поиска, также как и размер популяции N_P; F — параметр, управляющий усилением дифференциальных вариаций; C_r — параметр, управляющий вероятностью выбора мутированного значения. Параметры C_r и F влияют на скорость сходимости и робастность процесса поиска. Их оптимальные значения зависят от целевой функции, ограничений и размера популяции. В большинстве случаев значения этих параметров находятся методом проб и ошибок. Практические советы по выбору этих параметров можно найти в работе [2].

В классической схеме выбора метода ДЭ популяция следующего поколения выбирается следующим образом:

$$X_i^{(G+1)} = \begin{cases} U_i^{(G+1)}, & \text{если} \quad f\left(U_i^{(G+1)}\right) \leqslant f\left(X_i^{(G)}\right), \\ X_i^{(G)} & \text{иначе}. \end{cases} \tag{10}$$

Таким образом, каждый индивидуум следующей популяции сравнивается со своим дубликатом из текущей популяции. Тот индивидуум, который в бóльшей степени соответствует условиям оптимальности, переходит в следующее поколение. Заметим, что индивидуумы следующего поколения либо сохраняют свои свойства, либо становятся лучше по сравнению с их дубликатами в предыдущем поколении. Следует также отметить, что в схеме выбора метода ДЭ промежуточный (пробный) индивидуум не сравнивается со всеми индивидуумами текущей популяции, а только противопоставляется своему дубликату.

Ограничения. Под граничными значениями понимаются ограничения на переменные вида

$$L \leqslant X \leqslant H, \quad l_j \in L, \quad h_j \in H. \tag{11}$$

Необходимо, чтобы после репродуцирования новые значения переменных не выходили за допустимые пределы. Для этого переменные, значения которых не соответствуют граничным условиям, случайным образом возвращаются в допустимые пределы:

$$u_{ij}^{(G+1)} = \\ = \begin{cases} \operatorname{rand}_{ij}[0,1](h_j - l_j) + l_j, \text{ если } \left(u_{ij}^{(G+1)} < l_j\right) \vee \left(u_{ij}^{(G+1)} > h_j\right), \\ u_{ij}^{(G+1)} \quad \text{иначе}, \end{cases}$$

$$i = 1, \ldots, N_P, \quad j = 1, \ldots, n. \tag{12}$$

Существуют также другие, менее эффективные способы учета граничных значений, например решение уравнения (9) при изменении его параметров до тех пор, пока полученное решение не будет удовлетворять этим значениям.

В работе [3] для устранения ограничений используется метод штрафных функций. В отличие от методов устранения ограничений, в которых отсеиваются недопустимые решения, в методе штрафных функций из-за ограничений вводятся штрафы за удаление от допустимой области непосредственно в целевую функцию:

$$
\begin{gathered}
f_{\text{ш}}(X) = (f(X) + a)\prod_{i=1}^{m} c_i^{b_i}, \\
c_i = \begin{cases} 1 + s_i g_i(X), & \text{если} \quad g_i(X) > 0, \\ 1 \quad \text{иначе}, \end{cases} \\
s_i \geqslant 1, \quad b_i \geqslant 1, \\
\min f(X) + a > 0, \quad a = \text{const}.
\end{gathered} \tag{13}
$$

Путем изменения константы a можно достичь того, чтобы целевая функция принимала только неотрицательные значения. Даже если константа a принимает большие значения, это не влияет на процесс поиска. При помощи константы s_i вводят масштаб значений ограничительной функции. При помощи константы b_i модифицируют форму поверхности оптимизации. Когда величина функции при выходе ее из допустимой области незначительна, то следует увеличивать значения констант s_i и b_i. Обычно удовлетворительные результаты достигаются при значениях констант $s_i = 1$ и $b_i = 1$.

Данный метод требует введения дополнительных управляющих параметров, и, следовательно, необходимо подобрать значения таких параметров, при которых поиск происходит более эффективно. Как правило, это реализуется методом проб и ошибок, т.е. алгоритм оптимизации запускается несколько раз при различных значениях параметров. Очевидно, что такой подход не отличается эффективностью.

В работе [4] предложен оригинальный подход для решения проблемы ограничений — предложена модификация правила выбора (см. уравнение (10)), при которой нет необходимости использовать штрафные функции:

$$X_i^{(G+1)} = \\ = \begin{cases} U_i^{(G+1)}, & \text{если}\Big(\big(\forall j \in \{1,\ldots,m\}\colon g_j\big(U_i^{(G+1)}\big) \leqslant 0 \wedge \\ & \wedge\, g_j\big(X_i^{(G)}\big) \leqslant 0\big) \wedge \big(f\big(U_i^{(G+1)}\big) \leqslant f\big(X_i^{(G)}\big)\big)\Big) \vee \\ & \vee \Big(\big(\exists j \in \{1,\ldots,m\}\colon g_j\big(U_i^{(G+1)}\big) > 0\big) \wedge \\ & \wedge \big(\forall j \in \{1,\ldots,m\}\colon \max\big(g_j\big(U_i^{(G+1)}\big), 0\big) \leqslant \\ & \leqslant \max\big(g_j\big(X_i^{(G)}\big), 0\big)\big)\Big), \\ X_i^{(G)} & \text{иначе.} \end{cases} \qquad (14)$$

Таким образом, пробный вектор $U_i^{(G+1)}$ будет выбран, если он удовлетворяет всем ограничениям и обеспечивает меньшее значение, чем значение целевой функции, или если он обеспечивает значение, не превосходящее значение $X_i^{(G)}$, для всех функций ограничений.

Отметим, что в случае недопустимого решения значения целевой функции не вычисляются. Такой принцип обеспечивает быструю сходимость, что продемонстрировано на примерах в работе [4].

Выбор лучшего индивидуума основан на следующих принципах:

— если оба решения $X_i^{(G)}$ и $U_i^{(G+1)}$ допустимы, то предпочтение отдается решению с меньшим значением целевой функции;

— допустимое решение всегда предпочтительнее недопустимого;

— если оба решения недопустимы, то предпочтение отдается менее недопустимому решению.

Чтобы избежать явления стагнации [5, 6], при одинаковых характеристиках пробного и целевого векторов предпочтение отдается пробному вектору.

Целые и дискретные переменные. В канонической форме метод ДЭ — это метод оптимизации непрерывных переменных [2]. Однако в работе [1] представлена модификация метода ДЭ для целых и дискретных переменных. Рассмотрим оптимизацию целых переменных.

Во-первых, несмотря на то, что в методе ДЭ на промежуточных операциях используются непрерывные значения, для вычисления целевой функции используются целые переменные. Таким образом, имеем

$$f = f(y_i), \quad i = 1,\ldots,n,$$

где

$$y_i = \begin{cases} x_i & \text{для непрерывных переменных,} \\ \operatorname{int}(x_i) & \text{для целых переменных,} \end{cases} \\ x_i \in X. \qquad (15)$$

Функция $\mathrm{int}(.)$ преобразует непрерывные значения в целые отбрасыванием дробной части и используется только при вычислении целевой функции, при этом алгоритм продолжает работать с непрерывными значениями. Такой подход обеспечивает бóльшее разнообразие популяции и робастность алгоритма.

Во-вторых, в случае целых переменных инициализация популяции происходит согласно следующим начальным условиям:

$$P^{(0)} = x_{ij}^{(0)} = \mathrm{rand}_{ij}[0,1]\,(h_j - l_j + 1) + l_j,\ i = 1,\ldots,N_P,\ j = 1,\ldots,n. \tag{16}$$

Вместо уравнения (12) используется уравнение для целых переменных:

$$u_{ij}^{(G+1)} = \begin{cases} \mathrm{rand}_{ij}[0,1](h_j - l_j + 1) + l_j, \\ \qquad \text{если} \ \left(\mathrm{int}\left(u_{ij}^{(G+1)}\right) < l_j\right) \vee \left(\mathrm{int}\left(u_{ij}^{(G+1)}\right) > h_j\right), \\ u_{ij}^{(G+1)} \quad \text{иначе}, \end{cases} \tag{17}$$

$$i = 1,\ldots,N_P, \quad j = 1,\ldots,n.$$

Также легко могут быть обработаны дискретные переменные. Предположим, что множество дискретных переменных $X^{(d)}$ содержит l элементов:

$$X^{(d)} = x_i^{(d)}, \qquad x_i^{(d)} < x_{i+1}^{(d)}, \quad i = 1,\ldots,l. \tag{18}$$

Вместо дискретных значений $x_i^{(d)}$ в алгоритме используются их собственные индексы i. Теперь дискретные значения могут быть обработаны как целые переменные с граничными значениями $i \in [1,l]$. Для вычисления целевой функции используется непосредственно дискретное значение вместо его индекса. Другими словами, задача оптимизации дискретных переменных сводится к задаче оптимизации целых переменных, а дискретные значения используются только для вычисления целевой функции.

Результаты исследования. Для управляющих переменных выбраны значения $N_P = 40$, $F = 0{,}9$ и $C_r = 0{,}9$. Несмотря на то, что в постановке задачи не определены граничные значения конструкционных переменных, некоторые ограничения целесообразно представить в виде табл. 2. Остальные ограничения представим как штрафы в целевой функции (13).

Проблема была решена четырьмя различными методами, результаты представлены в табл. 3.

Таблица 2

Условия на нижней границе	Ограничение	Условия на верхней границе
Необходим по крайней мере один виток, чтобы сформировать пружину	$1 \leqslant x_1 \leqslant \dfrac{l_{\max}}{d_{\min}}$	Верхняя и нижняя поверхности ненагруженной пружины касаются друг друга
Ограничения g_3и g_5	$3d_{\min} \leqslant x_2 \leqslant D_{\max}$	Ограничение g_4
Ограничение g_3	$d_{\min} \leqslant x_3 \leqslant \dfrac{D_{\max}}{3}$	Ограничения g_4 и g_5

Таблица 3

Переменные и ограничения	Методы решения				Тип переменной
	1	2	3	4	
$x_1(N)$, ед.	10	9	9	9	целая
$x_2(D)$, мм	29,9898054	31,20898	31,1762394	31,0652414	непрерывная
$x_3(d)$, мм	7,1882	7,1882	7,1882	7,1882	дискретная
$g_1(X)$, Па	7,876853817	0,060331198	0,079914771	0,146315711	
$g_2(X)$, мм	223,99498	226,58578	226,73056	227,219256	
$g_3(X)$, мм	2,107692	2,1082	2,1082	2,1082	
$g_4(X)$, мм	46,21022	44,99102	45,02404	45,134784	
$g_5(X)$, ед.	1,1723	1,3417	1,3371	1,32170	
$g_6(X)$, мм	5,4643	5,4568	5,4585	5,46429	
$g_7(X)$, мм	0	0	0	$6{,}79656 \cdot 10^{-15}$	
$g_8(X)$, мм	0	0,44196	0,34036	$1{,}28909 \cdot 10^{-14}$	
$f(X)$, мм^3	359536,1515	312229,5723	311248,6371	307921,8791	

П р и м е ч а н и е: 1 — метод “ветвей и границ” с использованием последовательного квадратичного программирования; 2 — генетические алгоритмы; 3 — метагенетические алгоритмы; 4 — метод ДЭ

Заключение. При сравнении различных алгоритмов оптимизации основными критериями принято считать точность найденного решения и его оптимальность, а также время, затраченное на поиск этого решения. Эти два критерия в простой форме дают оценку таким характеристикам алгоритма, как его сходимость и сложность.

В задачах технического проектирования критерий времени маловажен, поэтому он не рассматривается. Тем не менее, следует заметить, что алгоритм ДЭ требует лишь несколько минут вычислений, в то время как для генетических алгоритмов 2, 3 и метода перебора 1 требуются часы.

Как видно из табл. 3, наиболее точные результаты получены методом ДЭ.

В заключение следует подчеркнуть, что в настоящее время метод ДЭ является одним из перспективных эвристических методов оптимизации среди эволюционных алгоритмов. Не уступая по скорости быстрым методам имитации отжига, он превосходит по точности генетические алгоритмы, оставаясь при этом легким в реализации и простым в использовании.

СПИСОК ЛИТЕРАТУРЫ

1. Lampinen J., Zelinka I. Mixed Integer–Discrete–Continuous Optimization by Differential Evolution. Part 1: The Optimization Method // Proc. of 5th International Conference on Soft Computing Mendel'99 (Brno, Czech Republic, 1999, June 9–12). – Brno University of Technology, 1999. – P. 71–76.
2. Storn R., Kenneth P. Differential Evolution — a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces // Technical Report TR–95-012. – ICSI, 1995.
3. Lampinen J., Zelinka I. Mixed Integer–Discrete–Continuous Optimization by Differential Evolution. Part 2: A Practical Example // Proc. of 5th International Conference on Soft Computing Mendel'99 (Brno, Czech Republic, 1999, June 9–12). – Brno University of Technology, 1999. – P. 77–81.
4. Lampinen J. Solving Problems Subject to Multiple Nonlinear Constraints by the Differential Evolution // Proc. of 7th International Conference of Soft Computing Mendel'2001 (Brno, Czech Republic, 2001, June 6–8). – Brno University of Technology, 2001. – P. 50–57.
5. Lampinen J., Zelinka I. On Stagnation of the Differential Evolution Algorithm // Proc. of International Conference on Soft Computing Mendel'2000 (Brno, Czech Republic, 2000, June 7–9). – Brno University of Technology, 2000. – P. 76–83.
6. Пупков К. А., Фалдин Н. В., Егупов Н. Д. Методы синтеза оптимальных систем автоматического управления. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2000. – 512 с.

Статья поступила в редакцию 25.09.2003

Константин Александрович Пупков родился в 1930 г., окончил в 1954 г. МВТУ им. Н.Э. Баумана. Д-р техн. наук, зав. кафедрой “Системы автоматического управления” МГТУ им. Н.Э. Баумана, действительный член РАЕН. Автор более 200 научных работ в области теории автоматического управления.

K.A. Pupkov (b. 1930) graduated from the Bauman Moscow Higher Technical School in 1954. D. Sc. (Eng.), professor, head of “Automatic Control Systems” department of the Bauman Moscow State Technical University, full member of Russian Academy of Natural Sciences. Author of over 200 publications in the field of theory of automatic control.

Виталий Андреевич Феоктистов родился в 1978 г., окончил в 2001 г. МГТУ им. Н.Э. Баумана. Аспирант кафедры “Системы автоматического управления” МГТУ им. Н.Э. Баумана. Автор 4 научных работ в области разработки и исследования эволюционных алгоритмов оптимизации.

V.I. Feoktistov (b. 1978) graduated from the Bauman Moscow State Technical University in 2001. Post-graduate of “Automatic Control Systems” department of the Bauman Moscow State Technical University. Author of 4 publications in the field of development and investigations of evolutionary algorithms of optimization.
