

УДК 519.71

А.А. Бессонов

Харьковский национальный университет радиоэлектроники, Харьков

## МНОГОКРИТЕРИАЛЬНАЯ НЕЙРОЭВОЛЮЦИОННАЯ ОПТИМИЗАЦИЯ НЕЛИНЕЙНЫХ ФУНКЦИЙ

Рассматривается возможность применения моделей нелинейных функций, полученных с помощью эволюционирующей радиально-базисной сети, для решения задачи многокритериальной оптимизации. Решение данной задачи заключается в поиске Парето-оптимального множества решений, которое предполагает многократное моделирование оптимизируемой системы при различных значениях варьируемых параметров. В реальных системах для этого обычно требуются значительные вычислительные затраты, в связи с чем возникает необходимость построения их быстродействующих моделей. Имитационное моделирование показало, что с помощью эволюционирующей радиально-базисной сети возможно получение достаточно точных моделей, использование которых позволяет определить множество Парето-оптимальных решений.

**Ключевые слова:** многокритериальная оптимизация, нейронная сеть, генетический алгоритм, нелинейная система, множество Парето, оптимальное решение.

### Введение

Задача многокритериальной оптимизации (МО), которую также часто называют мультикритериальной оптимизацией или векторной оптимизацией, заключается в нахождении такого вектора решений, удовлетворяющего определенным ограничениям, который давал бы приемлемые значения для всех целевых функций [1]. Следовательно, существует множество целевых функций (вектор целей), которые оптимизируются (минимизируются или максимизируются) одновременно. Эти цели часто вступают в противоречие друг с другом таким образом, что улучшение одной из них приводит к ухудшению другой, т.е. не существует единого оптимального решения, наилучшего по отношению ко всем целевым функциям. Вместо этого есть множество оптимальных решений задачи многоцелевой оптимизации, известное как Парето оптимальные решения или фронт Парето [2]. Понятие фронта Парето в области значений целевых функций в задаче МО обозначает набор таких решений, которые являются недоминируемыми по отношению друг к другу, но в то же время доминирующими над всеми остальными решениями в пространстве поиска. Это означает, что невозможно найти единое решение, которое бы превосходило все другие по отношению ко всем целям, т.е. переход между решениями, принадлежащими фронту Парето, не

может привести к улучшению всех целей одновременно. Следовательно, такой переход приведет к ухудшению не менее одной цели. Таким образом, каждое решение множества Парето включает в себе, по крайней мере, одну цель хуже, чем у другого решения из данного множества, хотя они оба превосходят все другие в оставшейся части пространства поиска. Математически данная задача может быть сформулирована следующим образом.

Требуется найти такой вектор

$$X^* = [x_1^*, x_2^*, \dots, x_n^*]^T,$$

который бы оптимизировал вектор целевых функций

$$F(X) = [f_1(X), f_2(X), \dots, f_k(X)]^T \quad (1)$$

при наличии  $m$  ограничений в виде неравенств

$$g_i(X) \leq 0, i = \overline{1, m}, \quad (2)$$

и  $p$  ограничений в виде равенств

$$h_j(X) = 0, j = \overline{1, p}, \quad (3)$$

где  $X^* \in \mathcal{R}^n$  – вектор решений;  $F(X) \in \mathcal{R}^k$  – вектор целевых функций, каждая из которых должна быть оптимизирована. Обычно полагают, что все целевые функции должны быть минимизированы. Многокритериальная минимизация на основе подхода Парето может быть осуществлена с помощью следующих определений.

**Доминирование по Парето.** Вектор

$$U = [u_1, u_2, \dots, u_n]^T \in \mathfrak{R}^k$$

доминирует над вектором  $V = [v_1, v_2, \dots, v_n]^T \in \mathfrak{R}^k$  (обозначается  $U \prec V$ ) тогда и только тогда, когда  $\forall i \in \{1, 2, \dots, k\}, u_i \leq v_i \wedge \exists j \in \{1, 2, \dots, k\} : u_j < v_j$ . Другими словами, существует как минимум одна компонента вектора  $u$  ( $u_j$ ), которая меньше, чем  $v_j$ , в то время как остальные компоненты вектора  $u$  меньше либо равны соответствующим компонентам вектора  $v$ .

**Оптимальность по Парето.** Точка  $X^* \in \Omega$  ( $\Omega$  некоторая область пространства  $\mathfrak{R}^n$ , удовлетворяющая условиям (2) – (3)) является Парето-оптимальной по отношению ко всем  $X \in \Omega$  тогда и только тогда, когда  $F(X^*) \prec F(X)$ , т.е. решение  $X^*$  является Парето-оптимальным, если не может быть найдено никакое другое решение, которое бы доминировало над  $X^*$  с учетом определения доминирования по Парето.

**Множество Парето.** Для данной задачи МО множеством Парето  $P^*$  называется такой набор векторов  $X \in \Omega$ , что не существует такого вектора  $X' \in \Omega$ , для которого бы выполнялось условие  $F(X') \prec F(X) \forall X \in P^*$ .

**Фронт Парето.** Для данной задачи МО фронтом Парето  $PF^*$  называется такой набор векторов значений целевых функций  $F(X) \in \mathfrak{R}^k$ , который получен с помощью векторов из множества Парето, т.е.

$$PF^* = \{F(X) = (f_1(X), f_2(X), \dots, f_k(X)) : X \in P^*\}.$$

Для нахождения фронта Парето в задачах МО широко используются эволюционные алгоритмы (ЭА), так как их свойства подходят для таких типов задач. Это обусловлено главным образом их параллельным или популяционным подходом к поиску решений, что позволяет устранить большинство трудностей и недостатков классических методов решения задач МО. В настоящее время наиболее часто используются следующие методы приближенного построения множества Парето на основе ЭА:

- VEGA (Vector Evaluated Genetic Algorithm) [3];
- FFGA (Fonseca and Fleming's Multiobjective Genetic Algorithm) [4];
- NPGA (Niche Pareto Genetic Algorithm) [5];
- SPEA (Strength Pareto Evolutionary Algorithm) [6];
- NSGA-II (Non-dominated Sorting Genetic Algorithm-II) [7].

В общем виде алгоритм поиска фронта Парето с помощью ЭА может быть записан следующим образом.

1. Сгенерировать начальную обучающую выборку, состоящую из векторов входных переменных

$X \in \Omega$ . Вычислить векторы значений целевых функций  $F(X)$  для всех  $X$ .

2. На основе обучающей выборки  $X \in \Omega$  и соответствующих значений  $F(X)$  построить модели всех целевых функций  $f_1(X), f_2(X), \dots, f_k(X)$ .

3. На основе полученных моделей  $f'_1(X), f'_2(X), \dots, f'_k(X)$  с помощью выбранного алгоритма определяется Парето-оптимальное множество  $PF^*$  решений задачи (1) – (3).

4. В точках полученного множества решений  $PF^*$  вычисляются точные значения функций  $f_1(X), f_2(X), \dots, f_k(X)$ . Если критерий останова не выполняется (получена требуемая точность моделей и построен фронт Парето либо осуществлено максимально допустимое число итераций), то все полученные значения добавляются в обучающую выборку и осуществляется возврат к шагу 2, на котором уточняются модели целевых функций.

Важным этапом данного алгоритма является получение моделей целевых функций. Наличие данных моделей обусловлено тем, что в реальных системах получение значений целевых функций является весьма трудоемкой задачей. При решении же задачи МО подобные вычисления необходимо проводить многократно. В связи с этим, точные и быстродействующие модели целевых функций в значительной мере ускоряют задачу получения Парето-оптимальных решений. Получить такие модели можно с помощью радиально базисных сетей (РБС), которые являются универсальными аппроксиматорами и при этом обладают простой структурой и отличаются высокой скоростью обучения.

## Структура РБС

РБС осуществляют аппроксимацию функции  $f(\mathbf{x})$  некоторой системой базисных функций (БФ) – нелинейных функций  $\varphi(\mathbf{x}, \mathbf{t})$ , зависящих от расстояния (радиального)  $r = \|\mathbf{x} - \mathbf{t}\|$ , где  $\mathbf{t}$  – вектор центров БФ. Представление нелинейной модели радиально-базисной сетью имеет вид

$$\hat{y}(k) = a_0 + \sum_{i=1}^N w_i \Phi_i(\mathbf{x}(k), \boldsymbol{\mu}, \sigma), \quad (4)$$

где  $a_0$  – смещение нейрона выходного слоя;  $w_i$  – вес связи  $i$ -го нейрона скрытого слоя с нейроном выходного слоя;  $N$  – число нейронов в скрытом слое;  $\Phi_i$  – базисная функция (БФ)  $i$ -го нейрона.

Выбор БФ играет важную роль при построении РБС, так как существенно влияет на сложность вычислений, а для упрощения обычно предполагается, что активационные функции нейронов одинаковы и выбраны экспертом. В РБС в качестве БФ могут быть использованы, например, следующие функции:

$$\Phi_i(x) = \exp\left\{-\frac{\|x - \mu_i\|^2}{\sigma_i^2}\right\}, \quad (5)$$

$$\Phi_i(x) = \left(1 - \frac{(x - \mu_i)^2}{\sigma_i^2}\right) e^{-\frac{(x - \mu_i)^2}{\sigma_i^2}}, \quad (6)$$

$$\Phi_i(x) = \exp\left\{-\frac{|x - \mu_i|}{\sigma_i}\right\} \quad (7)$$

$$\Phi_i(x) = \frac{2(x - \mu_i)}{\sigma_i} \exp\left\{-\frac{(x - \mu_i)^2}{\sigma_i^2}\right\}, \quad (8)$$

где  $\mu_i, \sigma_i$  – центры и радиусы базисных функций соответственно;  $\|\cdot\|$  – евклидова норма.

Обучение сети заключается в определении вектора

$$w(k) = (a_0(k), w_1(k), \mu_1^T(k), \sigma_1(k), \dots, w_N(k), \mu_N^T(k), \sigma_N(k))^T.$$

В настоящее время разработано и хорошо изучено достаточно большое количество алгоритмов настройки параметров сети, однако вопросы выбора ее оптимальной структуры остаются открытыми.

### Эволюционирующая РБС

При переходе от ИНС к ЭИНС для всех типов сетей используются общие эволюционные процедуры (инициализация популяции, оценка популяции, селекция, скрещивание, мутации), а различия заключаются лишь в способе кодирования структуры и параметров той или иной ИНС в виде хромосомы, содержащей определенный набор генов. Длина хромосомы обычно постоянна, а популяция, состоящая из некоторого количества особей, подвергается процессу эволюции с использованием операций скрещивания и мутаций.

Применение ЭА для обучения РБС позволяет одновременно решать задачи настройки параметров сети и определения ее оптимальной структуры. Более того, в отличие от большинства алгоритмов оптимизации, предназначенных для потактового решения задачи, ЭА оперируют с множеством решений – популяцией, что позволяет достичь глобального экстремума, не застревая в локальных.

Самым распространенным видом эволюционных алгоритмов являются генетические (ГА).

Классический ГА содержит следующие шаги:

1. Создание начальной популяции.
  - 1.1. Инициализация хромосомы каждой особи.
  - 1.2. Оценивание начальной популяции.
2. Этап эволюции – построение нового поколения.
  - 2.1. Отбор кандидатов на скрещивание (селекция).
  - 2.2. Скрещивание, т.е. порождение каждой парой отобранных кандидатов новых индивидов.
  - 2.3. Мутация.
  - 2.4. Оценивание новой популяции.

Критерием останова алгоритма обычно являются либо прохождение максимально допустимого количества поколений, либо достижение функцией приспособленности какой-либо особи некоторого заранее заданного порогового значения.

Подробно реализация ЭРБС и алгоритмы ее обучения рассмотрены в [8].

### Моделирование

Достаточно сложной задачей, требующей отдельного исследования, является задача тестирования алгоритмов многокритериальной оптимизации. Тестовые функции должны обладать рядом свойств, позволяющих с высокой точностью оценить эффективность разработанного алгоритма. В работе [9] предложена общая схема построения таких тестовых функций. Идея метода состоит в том, чтобы составить из некоторых базисных функций более сложные тестовые функции с хаотически расположенными глобальными оптимумами и несколькими случайно расположенными локальными оптимумами.

Составная тестовая функция при таком подходе вычисляется по формуле

$$F(x) = \sum_{i=1}^n w_i (f_i((x - o_i + o_{iold})\lambda_i M_i) + b_i) + b, \quad (9)$$

где  $w_i$  – вес каждой функции  $f_i(x)$ , вычисленный следующим образом:

$$w_i = \exp\left(-\sum_{k=1}^D (x_k - o_{ik} + o_{ikold})^2 / (2D\sigma_i^2)\right),$$

$$w_i = \begin{cases} w_i, & \text{если } w_i = \max(w); \\ w_i(1 - (\max(w))^n), & \text{иначе} \end{cases}$$

после вычисления всех весов осуществляется их нормализация

$$w_i = w_i / \sum_{i=1}^n w_i;$$

$n$  – число базисных функций;  $\sigma_i^2$  – радиус  $i$ -й базисной функции;  $\lambda_i$  – параметр, используемый для масштабирования функций;  $o_i$  – определяет положение локальных и глобального минимумов;  $b_i$  – определяет какой из оптимумов будет глобальным (наименьшее значение  $b_i$  соответствует глобальному оптимуму);  $D$  – размерность  $x$ ;  $M$  – ортогональная матрица поворота для каждой  $f_i(x)$ .

С помощью параметров  $o_i$  и  $b_i$  глобальный оптимум может быть расположен в любой заранее заданной точке.

В качестве базисных могут быть использованы, например, следующие тестовые функции:

– сферическая функция:

$$f(x) = \sum_{i=1}^D x_i^2, x \in [-100, 100]^D; \quad (10)$$

– функция Растригина

$$f(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10), x \in [-5, 5]^D; \quad (11)$$

– функция Вейерштрасса

$$f(x) = \sum_{i=1}^D \left( \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} [a^k \cos(\pi b^k)], a = 0.5, b = 3, \quad (12)$$

$$k_{\max} = 20, x \in [-0.5, 0.5]^D;$$

– функция Гриванка

$$f(x) = 1 + \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right), x \in [-600, 600]^D; \quad (13)$$

– функция Экли

$$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e, x \in [-32, 32]^D. \quad (14)$$

В данной работе для тестирования предложенного метода использовались в качестве нелинейной системы две функции от двух переменных, которые задавались следующим образом:

$$F_1(x_1, x_2), n = 10, D = 2:$$

$f_1, f_2, \dots, f_{10}$ : сферические функции

$$\sigma_1, \sigma_2, \dots, \sigma_{10} = [1, 1, \dots, 1];$$

$\lambda_1, \lambda_2, \dots, \lambda_{10}$  – равномерно распределенные в интервале  $[-5, 5]$  случайные величины;

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};$$

$$F_2(x_1, x_2):$$

$f_1, f_2, \dots, f_5$ : сферические функции (10);

$f_6, f_7, \dots, f_{10}$ : функции Экли (14);

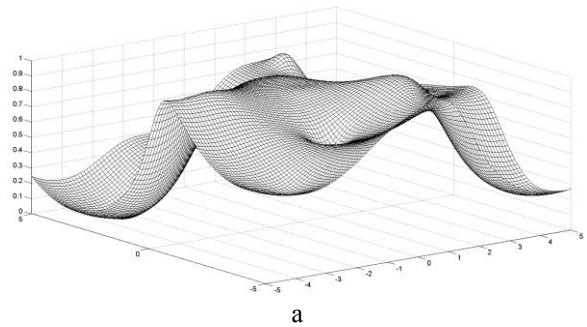
остальные параметры аналогичны.

Графики функций  $F_1(x_1, x_2)$  и  $F_2(x_1, x_2)$  приведены на рис. 1, а на рис. 2 показаны восстановленные с помощью ЭРБС поверхности. Обе модели были построены с помощью независимых ЭРБС. Начальная популяция состояла из 128 особей, а максимально допустимое количество нейронов было равно 30. На рис. 3 приведено изменение количества нейронов скрытого слоя для обеих сетей. На рис. 4 показано изменение значения фитнес-функций сетей победителей при построении каждой модели. Фронт Парето, построенный с использованием полученных моделей, приведен на рис. 5.

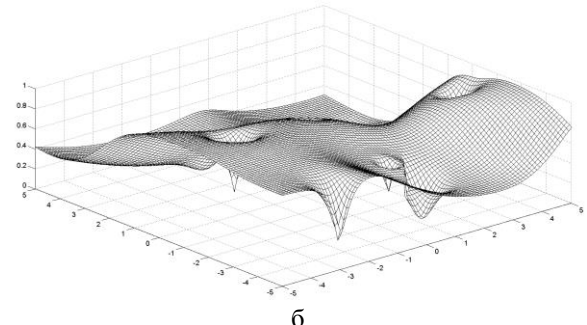
### Выводы

При решении задачи МО (1) – (3) с помощью достаточно хорошо изученных алгоритмов, основанных на понятии оптимальности по Парето, воз-

никает необходимость многократного вычисления значений целевых функций при различных значениях входных переменных, что в реальных системах является в ряде случаев весьма ресурсоемкой, а иногда и вовсе неразрешимой проблемой. В связи с этим возникает необходимость построения точных и быстродействующих моделей целевых функций. В данной работе для построения таких моделей предложено использовать эволюционирующую РБС.

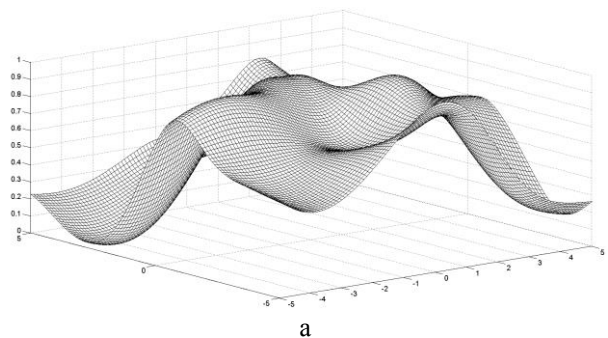


а

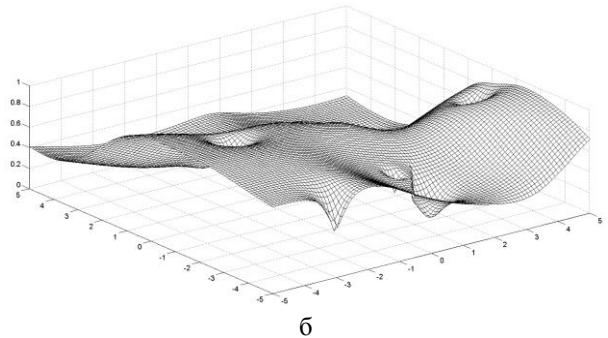


б

Рис. 1. Графики функций  $F_1(x_1, x_2)$  и  $F_2(x_1, x_2)$



а



б

Рис. 2. Восстановленные с помощью ЭРБС поверхности

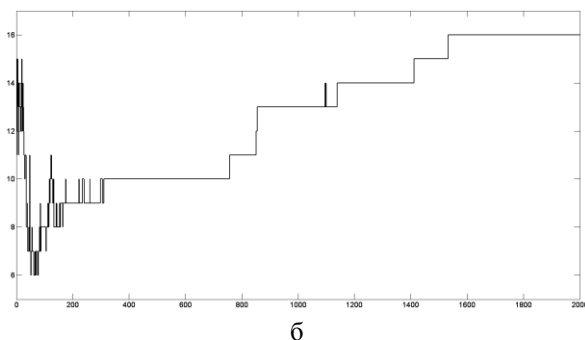
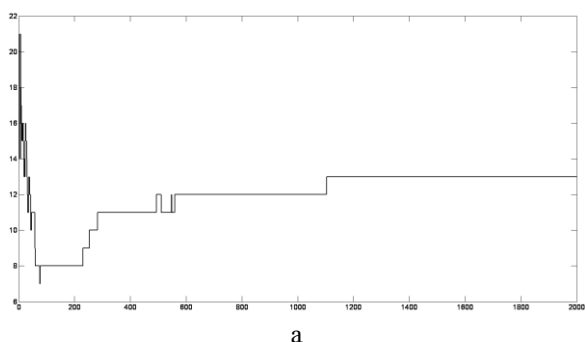


Рис. 3. Изменение количества нейронов скрытого слоя

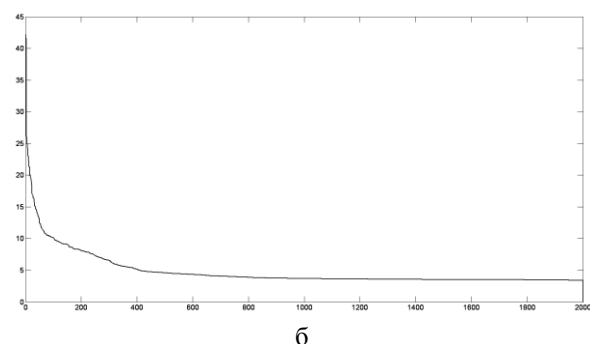
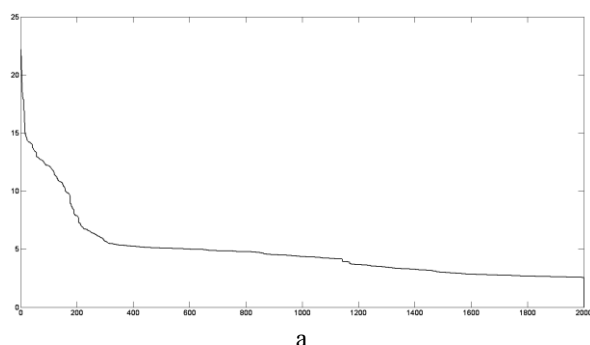


Рис. 4. Изменение значения фитнес-функций сетей

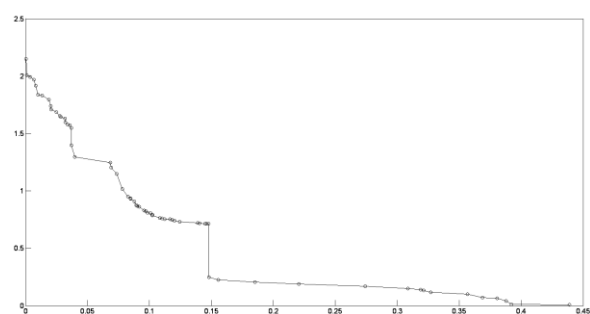


Рис. 5. Фронт Парето, построенный с использованием полученных моделей

Использование нейрореволюционного подхода, сочетающего ИНС и эволюционные вычисления, является достаточно универсальным и оказывается весьма эффективным в нестационарных условиях. Такой подход позволяет оперативно корректировать структуру и параметры сетей, а также алгоритмы их обучения в зависимости от изменения свойств как исследуемого объекта, так и окружающей среды, а выбор подходящей фитнес-функции обеспечивает робастность получаемых моделей.

Для моделирования предложенного подхода в работе используется алгоритм построения достаточно сложных целевых функций. Результаты имитационного моделирования в среде Matlab продемонстрировали эффективность применения ЭРБС в задачах МО.

## Список литературы

1. Coello Coello C.A. *Multi-objective optimization of trusses using genetic algorithms* / C.A. Coello Coello, A.D. Christiansen // *Computers & Structures*. – 2000. – V. 75. – P. 647-660.
2. Zitzler E. *Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach* / E. Zitzler, L. Thiele // *IEEE Trans. on Evolutionary Computation*. – 1999. – № 4, v. 3. – P. 257-271.
3. Schaffer J.D. *Multiple objective optimization with vector evaluated genetic algorithms* / J.D. Schaffer // *Proceedings of the 1st International Conference on Genetic Algorithms*. – 1985. – P. 93-100.
4. Fonseca C.M. *Genetic algorithm for multiobjective optimization, formulation, discussion and generalization* / C.M. Fonseca, P.J. Fleming // *Genetic Algorithms: Proceeding of the Fifth International Conference*. CA. – 1993. – P. 416-423.
5. Horn J.N. *A niched Pareto genetic algorithm for multiobjective optimization* / J.N. Horn, A.L. Nafpliotis, D.E. Goldberg // *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*. – IEEE Service Center, Piscataway, NJ, USA, Jun 1994. – P. 82-87.
6. Zitzler E. *An evolutionary algorithm for multiobjective optimization: The strength pareto approach* / E. Zitzler, L. Thiele // *Technical Report 43, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 1998*.
7. Deb K. *A Fast Elitist Multiobjective Genetic Algorithm: NSGA-II* / K. Deb, A. Pratap, S. Agarwal, T. Meyarivan // *IEEE Transactions on Evolutionary Computation*. – 2002. – № 2, v. 6. – P. 182-197.
8. Maillard E.P. *RBF neural network, basis functions and genetic algorithm* / E.P. Maillard, D. Gueriot // *Proc. Int. Conf. Neural Networks, Houston, TX*. – 1997. – V.4. – P. 2187-2192.
9. Liang J.J. *Novel Composition Test Functions for Numerical Global Optimization* / J.J. Liang, P.N. Suganthan, K. Deb // *IEEE Swarm Intelligence Symposium*. – 2005. – P. 68-75.

Поступила в редколлегию 24.10.2012

**Рецензент:** д-р техн. наук, проф. О.Г. Руденко, Харьковский национальный университет радиоэлектроники, Харьков.

**БАГАТОКРИТЕРІАЛЬНА НЕЙРОЕВОЛЮЦІЙНА ОПТИМІЗАЦІЯ НЕЛІНІЙНИХ ФУНКЦІЙ**

О.О. Безсонов

Розглядається можливість застосування моделей нелінійних систем, отриманих за допомогою еволюціонуючої радіально-базисної мережі, для вирішення задачі багатокритеріальної оптимізації, яка полягає в пошуку Парето-оптимальної множини рішень. Це передбачає багаторазове моделювання системи, що оптимізується, при різних значеннях варіюваних параметрів. У реальних системах для цього зазвичай потрібні значні обчислювальні витрати, у зв'язку з чим виникає необхідність побудови їх швидкодіючих моделей. Імітаційне моделювання довело, що за допомогою еволюціонуючої радіально-базисної мережі можливе отримання досить точних моделей, використання яких дозволяє визначити множину Парето-оптимальних рішень.

**Ключові слова:** багатокритеріальна оптимізація, нейронна мережа, генетичний алгоритм, нелінійна система, множина Парето, оптимальне рішення.

**MULTIOBJECTIVE NEUROEVOLUTIONARY OPTIMIZATION OF NONLINEAR FUNCTIONS**

O.O. Bezsonov

In this paper it is considered the possibility to utilize models of nonlinear systems obtained by evolving radial base network for solving multi-criteria optimization task. The solution to this problem is to find the Pareto-optimal set of solutions, which involves multiple simulation of optimized system for different values of variable parameters. In real systems it is usually requires significant computational costs what raises the need to construct their high-speed models. Simulation modeling showed that quite accurate models may be received by evolving radial basic network, what allow determining the set of Pareto-optimal solutions.

**Keywords:** multiobjective optimization, neural network, genetic algorithm, nonlinear system, Pareto set, optimal solution.