

УДК 004.023, 629.7.03

## МЕТОД МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ НА ОСНОВЕ ПРИБЛИЖЕННЫХ МОДЕЛЕЙ ИССЛЕДУЕМОГО ОБЪЕКТА

Ю. А. Зеленков<sup>1</sup>

Рассматривается метод многокритериальной оптимизации по алгоритму NSGA-II на основе приближенных моделей исследуемого объекта. Приближенные модели строятся на базе нейронных сетей радиального базиса (RBF-сетей), параметры которых определяются при помощи эволюционного алгоритма. В качестве примера рассматривается многокритериальная оптимизация параметров рабочего процесса газотурбинного двигателя.

**Ключевые слова:** приближенные модели, RBF-сети, многокритериальная оптимизация, расчет параметров ГТД.

**1. Введение.** В машиностроении инженерные вычисления являются основным способом снижения затрат на проектирование новой продукции, поскольку позволяют избежать длительных испытаний при доводке конструкции на натуральных образцах [1]. Существующие методы оптимизации [2] позволяют расчетным путем найти наиболее эффективное сочетание параметров изделия прежде, чем начинать изготовление опытных экземпляров. Особенно важным является использование многокритериальной оптимизации [3], что ведет, однако, к значительному увеличению числа выполняемых расчетов. Очень часто связи между целевыми функциями и независимыми переменными описываются системами уравнений в частных производных или интегро-дифференциальными уравнениями, решение которых можно получить только численными методами, а также эмпирическими зависимостями в виде таблиц, причем по своей природе эти зависимости имеют мультидисциплинарный характер.

Применяемые в современной практике расчеты, базирующиеся на методе конечных элементов (МКЭ), требуют значительных затрат машинного времени. Согласно экспериментальным результатам, полученным на НПО «Сатурн», для расчета газотурбинного двигателя (ГТД) на стационарном режиме (при постоянной частоте вращения ротора) требуется выполнение  $4.6 \times 10^{17}$  операций с плавающей точкой. На суперкомпьютере производительностью 0.6 Тфлопс данный расчет может быть выполнен за 14 дней при условии отсутствия потерь при распараллеливании задач. При многокритериальной оптимизации такие вычисления необходимо повторить 1000 и более раз. Таким образом, весьма актуальным является использование приближенных моделей оптимизируемых конструкций, позволяющих значительно сократить требуемые объемы вычислений.

**2. Постановка задачи.** Без потери общности будем рассматривать задачу многокритериальной минимизации с  $m$  независимыми переменными,  $n$  целями,  $p$  ограничениями в виде неравенств и  $q$  ограничениями в виде равенств [3]:

$$\text{минимизировать } \mathbf{f}(\mathbf{x}) \quad \text{при условии } \mathbf{g}(\mathbf{x}) \geq 0, \quad \mathbf{h}(\mathbf{x}) = 0, \quad (1)$$

где  $\mathbf{x} = (x_1, \dots, x_m) \in X$  является вектором решений (независимых переменных),  $X$  — пространство параметров,  $\mathbf{f}(\mathbf{x})^T = [f_1(\mathbf{x}), \dots, f_n(\mathbf{x})]$  — цели,  $\mathbf{g}(\mathbf{x})^T = [g_1(\mathbf{x}), \dots, g_p(\mathbf{x})]$  — ограничения в виде неравенств,  $\mathbf{h}(\mathbf{x})^T = [h_1(\mathbf{x}), \dots, h_q(\mathbf{x})]$  — ограничения в виде равенств. Вектор решений  $\mathbf{a} \in X$  является доминирующим над вектором  $\mathbf{b} \in X$  (обозначается как  $\mathbf{a} \prec \mathbf{b}$ ), если выполняется условие

$$\forall i \in \{1, \dots, n\} : f_i(\mathbf{a}) \leq f_i(\mathbf{b}) \wedge \exists j \in \{1, \dots, n\} : f_j(\mathbf{a}) < f_j(\mathbf{b}).$$

Вектор  $\mathbf{a}$  называют недоминируемым на множестве  $X' \subseteq X$ , если в  $X'$  нет вектора, доминирующего над  $\mathbf{a}$ . Множество решений  $X'$ , для которого выполняется условие

$$\forall \mathbf{a}' \in X' : \neg \exists \mathbf{a} \in X : \mathbf{a} \prec \mathbf{a}' \wedge \|\mathbf{a} - \mathbf{a}'\| < \varepsilon \wedge \|f(\mathbf{a}) - f(\mathbf{a}')\| < \delta,$$

где  $\|\dots\|$  — метрика расстояния и  $\varepsilon > 0$ ,  $\delta > 0$ , называется локальным Парето-оптимальным множеством.  $X'$  является глобальным Парето-оптимальным множеством, если  $\forall \mathbf{a}' \in X' : \neg \exists \mathbf{a} \in X : \mathbf{a} \prec \mathbf{a}'$  [4].

<sup>1</sup> Научно-производственное общество «Сатурн», просп. Ленина, 163, 152903, г. Рыбинск, Ярославская обл.; директор по информационным технологиям, e-mail: yuri.zelenkov@npo-saturn.ru

Таким образом, задача многокритериальной оптимизации является задачей поиска глобального Парето-оптимального множества решений. В случае проектирования авиационного ГТД данное множество представляется эксперту, который выбирает один из вариантов конструкции для дальнейшей проработки.

Анализ даже упрощенных методов термогазодинамического расчета одноконтурного ГТД [5] показывает, что на определение рабочего процесса и, следовательно, конструктивного облика двигателя влияют более 30 параметров (независимых переменных). При этом зависимости, связывающие целевые и независимые переменные, являются нелинейными, и невозможно гарантировать, что они являются дифференцируемыми функциями. На сегодняшний день известен ряд методов многокритериальной оптимизации, опирающихся на нелинейное программирование [6] и генетические алгоритмы [7], которые используются в том числе и при проектировании газотурбинных двигателей [8, 9]. Один из наиболее эффективных алгоритмов многокритериальной оптимизации с ограничениями — генетический алгоритм NSGA-II [10]. Особенностью данного алгоритма является то, что на каждом шаге вычислений генерируется новая популяция из  $N$  решений, для каждого из которых должны быть вычислены функции  $f(x)$ ,  $g(x)$  и  $h(x)$ . Типичной является популяция из 100 решений, которая эволюционирует в течение 500 поколений. Нетрудно оценить, что в этом случае необходимо 50 000 вычислений функций  $f(x)$ ,  $g(x)$ ,  $h(x)$ . Таким образом, исходя из практических соображений для того, чтобы сократить затраты времени до разумных пределов, необходимо предложить способ нахождения Парето-оптимального набора решений не более чем за 500 вычислений выражений точных моделей исследуемых зависимостей. Для достижения этой цели предлагается использовать подход, основанный на использовании вместо зависимостей (1) их приближенных моделей.

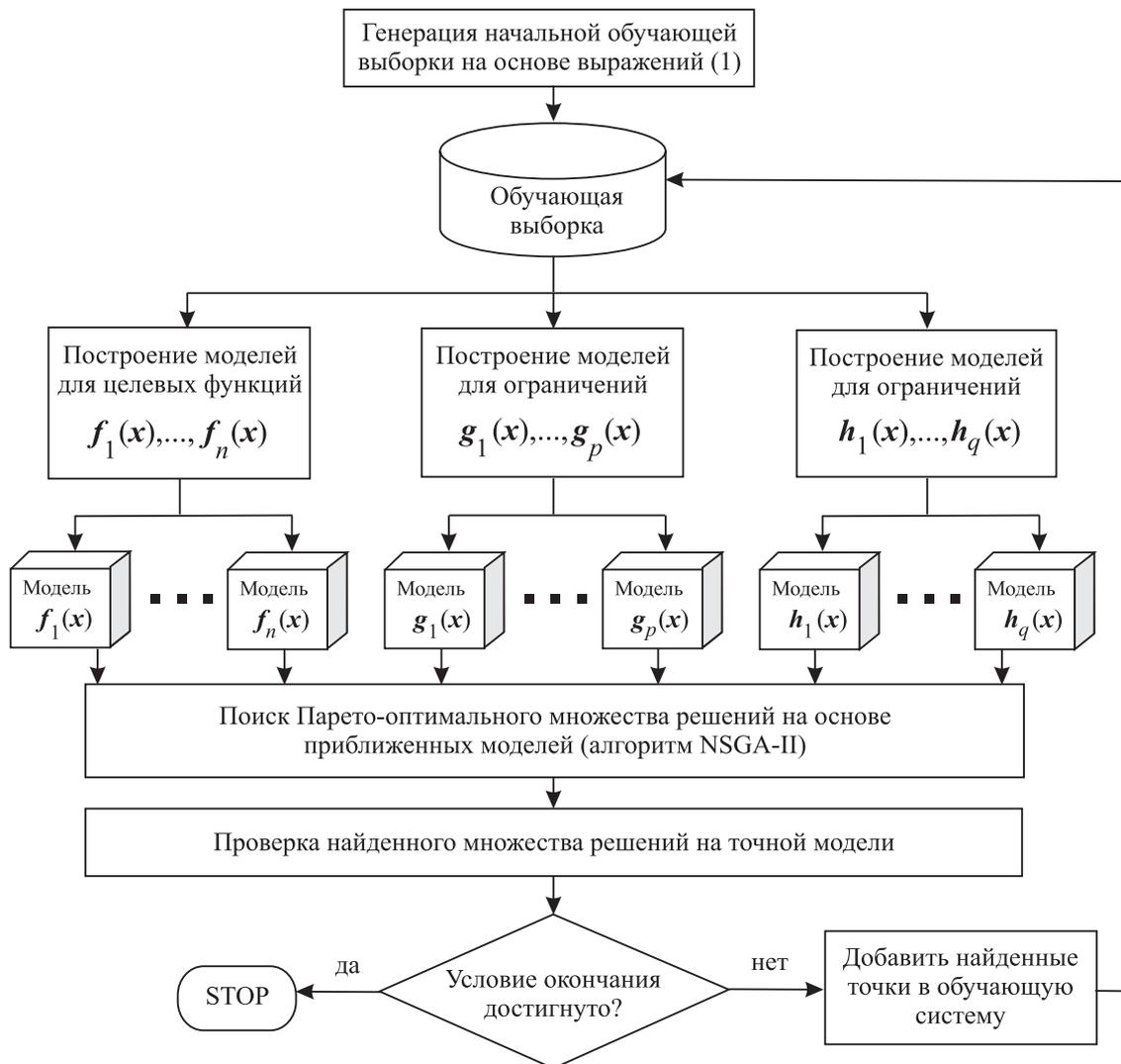


Рис. 1. Алгоритм метода многокритериальной оптимизации

**3. Метод многокритериальной оптимизации.** Предлагаемый алгоритм вычислений спроектирован исходя из следующих соображений. Как уже упоминалось выше, исследуемые зависимости носят междисциплинарный характер, невозможно гарантировать дифференцируемость или выпуклость этих функций. Кроме того, при разработке новых изделий никогда не ставится задача найти наилучшие параметры нового продукта (т.е. глобальное Парето-оптимальное множество), а все усилия направлены лишь на обеспечение заданных технических требований. Очень часто выбирается решение с менее эффективными параметрами, но обеспечивающее большую устойчивость к отклонениям, которые неизбежно возникают в процессе производства. В-третьих, избыточно большое Парето-оптимальное множество требует значительных затрат времени и ресурсов на анализ всех альтернативных решений, вполне допустимым является наличие 15–20 вариантов проектируемого изделия. Поэтому в предлагаемом методе многокритериальной оптимизации используются эвристические подходы (эволюционные и генетические алгоритмы). Описываемый метод (рис. 1) состоит из следующих шагов.

1. Генерируется начальная обучающая выборка  $\mathbf{x}_s$  небольшого объема  $s \in X$  на основе одного из методов планирования эксперимента. Вычисляются векторы значений целевых функций  $\mathbf{f}(\mathbf{x}_s)$  и ограничений  $\mathbf{g}(\mathbf{x}_s)$  и  $\mathbf{h}(\mathbf{x}_s)$  во всех полученных точках.

2. На основе обучающей выборки  $\mathbf{x}_s$  и соответствующих значений  $\mathbf{f}(\mathbf{x}_s)$ ,  $\mathbf{g}(\mathbf{x}_s)$  и  $\mathbf{h}(\mathbf{x}_s)$  строятся приближенные модели  $\tilde{\mathbf{f}}(\mathbf{x})$ ,  $\tilde{\mathbf{g}}(\mathbf{x})$  и  $\tilde{\mathbf{h}}(\mathbf{x})$  всех исследуемых зависимостей. Предлагаемый метод построения приближенных моделей рассматривается ниже.

3. На основе полученных приближенных моделей  $\tilde{\mathbf{f}}(\mathbf{x})$ ,  $\tilde{\mathbf{g}}(\mathbf{x})$  и  $\tilde{\mathbf{h}}(\mathbf{x})$  при помощи алгоритма NSGA-II находится вектор  $\mathbf{x}_{\text{opt}}$ , который определяет Парето-оптимальное множество решений задачи (1).

4. В точках полученного таким образом множества решений  $\mathbf{x}_{\text{opt}}$  вычисляются точные значения функций  $\mathbf{f}(\mathbf{x}_{\text{opt}})$ ,  $\mathbf{g}(\mathbf{x}_{\text{opt}})$  и  $\mathbf{h}(\mathbf{x}_{\text{opt}})$ . Если условие окончания вычислений не выполняется, то все значения, полученные на точных моделях, добавляются в обучающую выборку:

$$\mathbf{x}_s = \mathbf{x}_s + \mathbf{x}_{\text{opt}}, \quad \mathbf{f}(\mathbf{x}_s) = \mathbf{f}(\mathbf{x}_s) + \mathbf{f}(\mathbf{x}_{\text{opt}}), \quad \mathbf{g}(\mathbf{x}_s) = \mathbf{g}(\mathbf{x}_s) + \mathbf{g}(\mathbf{x}_{\text{opt}}), \quad \mathbf{h}(\mathbf{x}_s) = \mathbf{h}(\mathbf{x}_s) + \mathbf{h}(\mathbf{x}_{\text{opt}})$$

и осуществляется возврат к шагу 2, на котором вновь строятся приближенные модели.

Определим следующие условия окончания вычислений:

— суммарная относительная погрешность  $e$  построенных моделей достигает заданного минимума:

$$e = \frac{1}{k(n+p+q)} \sum_{j=1}^{n+p+q} \sqrt{\sum_{i=1}^k \left[ \frac{M_{ij}(x) - F_{ij}(x)}{F_{ij}(x)} \right]^2} \leq \varepsilon, \quad (2)$$

где  $k$  — количество решений в найденном Парето-оптимальном множестве;  $M_{ij}(x)$  — значение одной из функций  $\mathbf{f}(\mathbf{x})$ ,  $\mathbf{g}(\mathbf{x})$  или  $\mathbf{h}(\mathbf{x})$ , найденное на основе ее приближенной модели;  $F_{ij}(x)$  — значение той же функции, найденное на основе точной модели, и  $\varepsilon$  — достаточно малое положительное число. Выполнение данного условия означает, что качество построенных приближенных моделей таково, что позволяет их использовать вместо точных;

— нахождение одного или нескольких векторов  $\mathbf{f}(\mathbf{x})$ , удовлетворяющих заранее определенным требованиям  $\mathbf{f}(\mathbf{x}) \leq \mathbf{f}_{\text{goal}}$  при соблюдении ограничений  $\mathbf{g}(\mathbf{x}) \geq 0$  и  $\mathbf{h}(\mathbf{x}) = 0$ , где  $\mathbf{f}_{\text{goal}}$  — заданные экспертом значения целевых функций, достаточные для обеспечения необходимых характеристик проектируемого изделия;

— превышение допустимого количества точных вычислений моделей;

— превышение допустимого времени вычислений.

**4. Метод построения приближенной модели исследуемого объекта.** Ключевым вопросом успеха предлагаемого алгоритма является выбор эффективного способа построения приближенной модели. В зарубежной литературе такие модели называют “поверхности отклика” (response surface model — RSM). В частности, для построения приближенных функциональных зависимостей используется метод группового учета аргументов [11, 12], многослойные перцептроны и другие модели. В данной статье рассматривается использование для моделирования зависимостей (1) искусственных нейронных сетей радиального базиса (RBF-сетей), получаемых с помощью эволюционных алгоритмов [13].

Известно несколько таких методов, в частности достаточно общий, описанный в работе [14]. Недостатком данного метода является избыточность в описании сети (вводятся отдельные матрицы для описания весов, связей и вектор для описания нейронов). Упрощенная версия этого способа описания сети рассмотрена в [15]. Согласно данному методу объектом эволюции является популяция нейронных сетей. Кроме того, известны работы (в частности [16]), которые ограничиваются рассмотрением только RBF-сетей, что

позволяет перейти к рассмотрению эволюционирования популяции нейронов, которые затем объединяются в сеть. Однако последний алгоритм применим только для генерации сетей для классификации образов, поскольку он предполагает знание центров классов исследуемых объектов.

Основная особенность метода эволюционных алгоритмов [17], отличающая его от аналогичного метода генетических алгоритмов, — отказ от использования операции кроссовера. В работе [18] на основе анализа многих источников сделан вывод, что для задачи генерирования нейронных сетей эволюционные алгоритмы являются более эффективным методом, поскольку операция кроссовера часто приводит к ухудшению приспособленности потомков.

Рассмотрим нейронную сеть с активационными функциями радиального базиса так, как это предложено в [19] (рис. 2). Функция активации нейрона скрытого слоя имеет вид

$$y_i = \varphi(\|\mathbf{x} - \mathbf{c}_i\|) = \exp\left[-\frac{\|\mathbf{x} - \mathbf{c}_i\|}{2\sigma_i^2}\right],$$

где  $\|\mathbf{x} - \mathbf{c}_i\| = \sqrt{\sum_{j=1}^N (x_j - c_{ij})^2}$  — евклидово расстояние

между вектором входных сигналов  $\mathbf{x} = (x_1, \dots, x_N)$  и центром  $i$ -го нейрона  $\mathbf{c}_i = (c_{i1}, \dots, c_{iN})$ ,  $i = 1, \dots, L$ ;  $L$  — число нейронов в скрытом слое;  $N$  — число нейронов во входном слое;  $\mathbf{c}_i$ ,  $\sigma_i$  — параметры радиальной базисной функции  $i$ -го нейрона. Сигнал нейрона выходного слоя определяется взвешенным суммированием выходов нейронов скрытого слоя  $f_k = \sum_{i=1}^L w_i y_i$ , где  $w_i$  — вес связи от  $i$ -го нейрона скрытого слоя к нейрону выходного слоя.

Введем обозначения:  $\mathbf{z} = (z_1, \dots, z_p)^T$  — вектор ожидаемых значений функции ( $p$  — количество обучающих выборок),  $\mathbf{w} = (w_1, \dots, w_L)^T$  — вектор весов,  $G$  — радиальная матрица, которая имеет вид

$$G = \begin{bmatrix} \varphi(\|x_1 - c_1\|) & \varphi(\|x_1 - c_2\|) & \dots & \varphi(\|x_1 - c_L\|) \\ \varphi(\|x_2 - c_1\|) & \varphi(\|x_2 - c_2\|) & \dots & \varphi(\|x_2 - c_L\|) \\ \dots & \dots & \dots & \dots \\ \varphi(\|x_p - c_1\|) & \varphi(\|x_p - c_2\|) & \dots & \varphi(\|x_p - c_L\|) \end{bmatrix}.$$

Тогда вектор весов может быть найден по формуле

$$\mathbf{w} = G^+ \mathbf{z}, \tag{3}$$

где  $G^+ = (G^T G)^{-1} G^T$  — псевдоинверсия прямоугольной матрицы  $G$ .

Таким образом,  $i$ -й нейрон скрытого слоя полностью может быть описан строкой из  $(N + 2)$  действительных чисел, которая содержит вектор  $\mathbf{c}_i = (c_{i1}, \dots, c_{iN})$ , величину  $\sigma_i$  и значение  $w_i$ . Следовательно, для описания всей сети целиком необходима матрица  $R$  размером  $L \times (N + 2)$ . Однако, поскольку в рассматриваемом методе используется самоадаптивный способ настройки весов, в описании нейрона необходимо добавить матрицу  $\eta$  такого же размера, содержащую вариации (стратегические параметры эволюционного алгоритма), использование которых будет объяснено позже.

Предлагаемый эволюционный алгоритм представлен на рис. 3. В начальной популяции все параметры в описании сети инициализируются случайными значениями из интервала  $(-1, 0; 1, 0)$ . Производится вычисление приспособленности всех особей популяции по формуле:

$$e_m = \frac{1}{T} \sqrt{\sum_{t=1}^T (Y(t) - Z(t))^2}, \tag{4}$$

где  $T$  — количество образцов в обучающей выборке,  $Y(t)$  и  $Z(t)$  — ожидаемое и фактическое значения на выходе сети. Механизм отбора особи основан на ее ранге. Пусть  $K$  особей отсортированы по убыванию

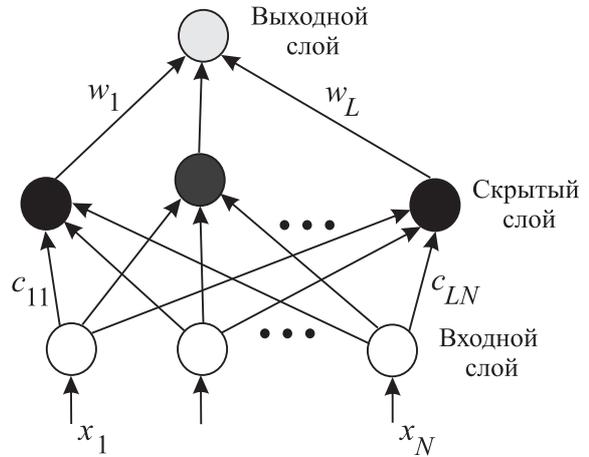


Рис. 2. Нейронная сеть радиального базиса (RBF-сеть)

функции (4) и им присвоены номера  $0, 1, \dots, K - 1$ . Тогда особь с номером  $(K - j)$  может быть отобрана для мутации с вероятностью  $p(K - j) = j \left( \sum_{k=1}^K k \right)^{-1}$ .

Перед началом мутации случайно выбирается целое число  $n$  из интервала  $(1, L)$ , определяющее номер нейрона, к которому будет применяться операция мутации. К данному нейрону последовательно применяются следующие операции мутации.

1. *Модификация параметров функции активации.* Используется гауссовская мутация, согласно которой новые значения строки матрицы  $R$  для данного нейрона вычисляются по формулам

$$\eta'_{nj} = \eta_{nj} \exp [\tau' N(0, 1) + \tau N_j(0, 1)];$$

$$R'_{nj} = R_{nj} + \eta'_{nj} N_j(0, 1),$$

$N(0, 1)$  — случайное число, подчиняющееся нормальному распределению со средним значением 0 и вариацией 1;  $N_j(0, 1)$  означает, что случайное число генерируется для каждого  $j$ -го элемента вектора;  $\tau = \left( \sqrt{2\sqrt{N}} \right)^{-1}$ ;  $\tau' = \left( \sqrt{2N} \right)^{-1}$ . После модификации параметров  $n$ -го нейрона уточняются веса по формуле (3) и вычисляется приспособленность полученной сети. Если она улучшается, полученный потомок помещается в новую популяцию, прочие мутации не производятся. В противном случае строкам  $R_n$  и  $\eta_n$  возвращаются старые значения и производится попытка выполнить следующую мутацию.

2. *Удаление нейрона.* Эта операция выполняется в случае неуспеха предыдущей мутации. Выбранный нейрон удаляется, по формуле (3) вычисляются весовые коэффициенты, оценивается приспособленность сети; если она улучшается, то полученный потомок копируется в новую популяцию. В противном случае применяется мутация добавления нейрона.

3. *Добавление нейрона.* Все параметры добавляемого нейрона инициализируются случайными значениями из интервала  $(-1.0; 1.0)$ , по формуле (3) вычисляются весовые коэффициенты. В случае улучшения приспособленности сети полученный потомок копируется в новую популяцию.

Если ни одна из мутаций не была успешной, то особь копируется в популяцию следующего поколения без изменений. Отметим, что в данном методе используется так называемый “жадный” алгоритм — попытка удаления нейрона всегда производится до попытки его добавить. Это обеспечивает получение более компактных сетей. Кроме того, используется принцип элитизма — лучшая особь текущей популяции копируется в новую без изменений.

5. **Оценка эффективности метода построения приближенных моделей.** Для оценки эффективности предложенного алгоритма рассмотрим исследованную в [20] задачу аппроксимации функции

$$d(x, y) = 3(1 - x)^2 \exp(-x^2 - (y + 1)^2) - 10 \left( \frac{x}{5} - x^3 - y^5 \right) \exp(-x^2 - y^2) - \frac{1}{2} \exp(-(x + 1)^2 - y^2) \quad (5)$$

при изменении переменных в пределах  $-3 \leq x \leq 3$  и  $-3 \leq y \leq 3$ . На основе обучающей выборки из 625 групп данных  $([x, y], d)$ , сгенерированных при равномерном распределении переменных  $x$  и  $y$  в областях их определения, в работе [20] построена сеть со структурой 2-36-1 (2 входных нейрона, 36 радиальных нейронов гауссовского типа и один выходной линейный нейрон). Применялся гибридный алгоритм обучения, в результате максимальная погрешность аппроксимации после 200 итераций составила 0.06. Согласно

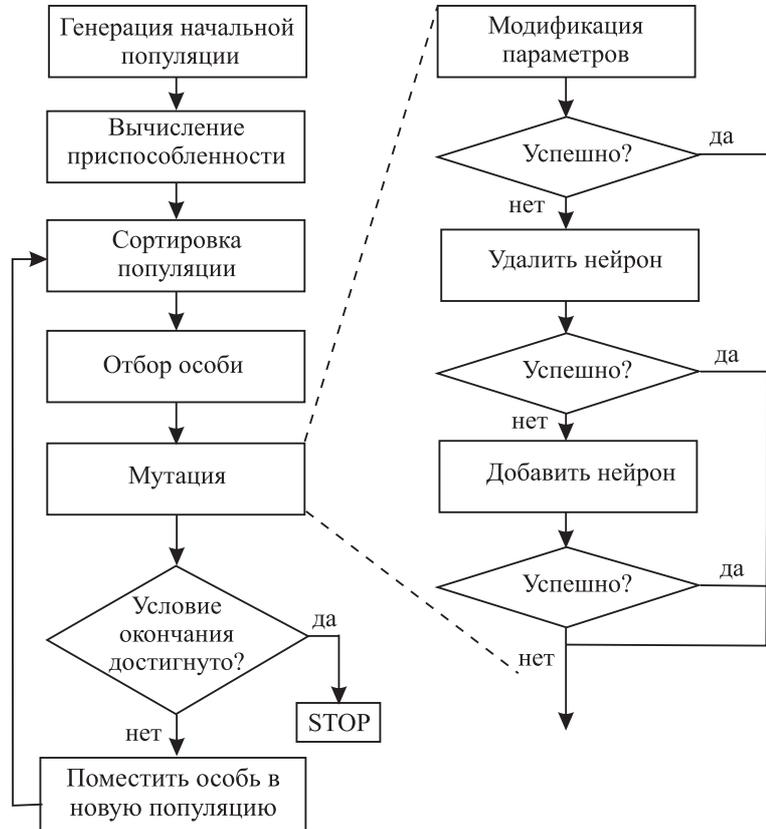


Рис. 3. Эволюционный алгоритм построения нейронной сети радиального базиса

предложенному здесь методу, на основе той же обучающей выборки за 20 поколений была сгенерирована нейронная сеть с 26 радиальными нейронами, погрешность аппроксимации которой имеет величину 0.02. График аппроксимируемой функции представлен на рис. 4, погрешность ее приближения на основе предложенного здесь метода — на рис. 5. Таким образом, предлагаемый метод генерации радиальных нейронных сетей позволяет значительно сократить время вычислений и обеспечивает получение более эффективных сетей (с меньшим числом нейронов и меньшей погрешностью) по сравнению с традиционным способом.

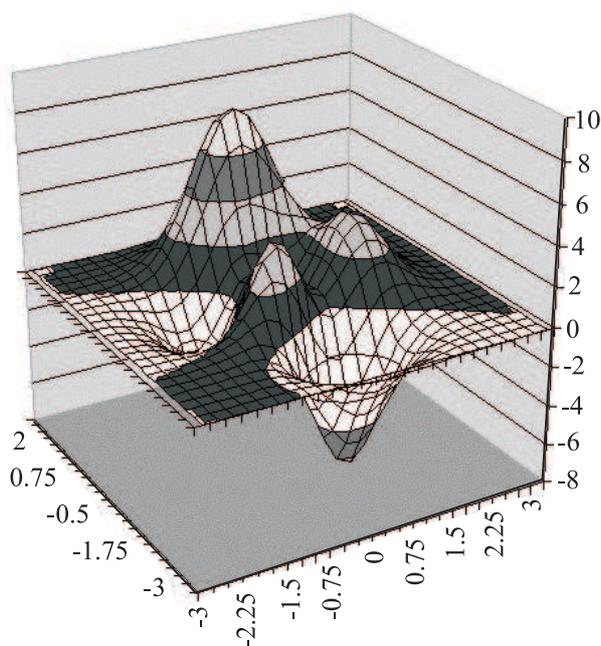


Рис. 4. График тестовой функции (5)

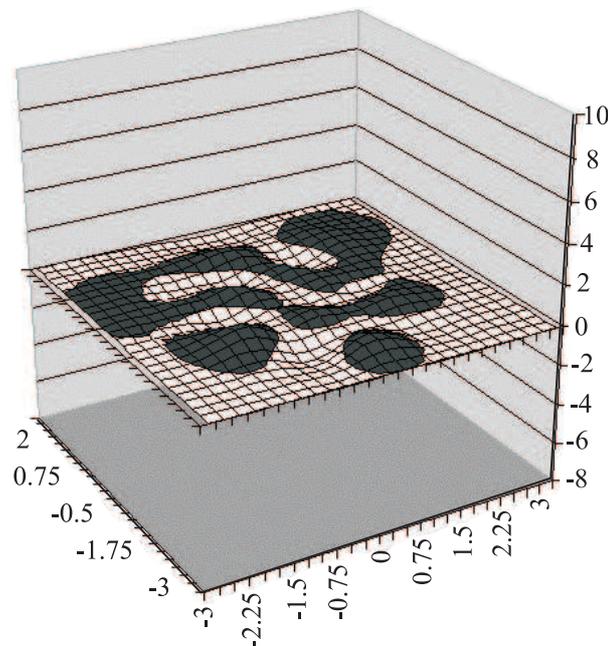


Рис. 5. Погрешность приближения тестовой функции (5)

Приближенные модели функции (5) были также построены на основе широко известных и используемых на практике методов, таких как многослойный перцептрон, сеть каскадной корреляции и метод группового учета аргументов. При этом общая выборка из 625 записей случайным образом была разделена на обучающую (90% записей) и тестовую (10% записей). Построение модели производилось на обучающей выборке, затем ее качество проверялось на тестовой выборке. Полученные значения среднеквадратичной погрешности (4) приведены в табл. 1. Предложенный здесь метод построения приближенных моделей показал наилучшие результаты.

**6. Математическая модель одноконтурного ГТД.** Для проверки эффективности предложенного способа оптимизации проведем исследования на упрощенной математической модели ГТД, которая требует гораздо меньших затрат времени на вычисление, чем МКЭ-модель, но, в то же время, обладает всеми особенностями реально используемых на практике функций (1). Для создания такой модели воспользуемся результатами работы [21], где приводятся экспериментальные зависимости, которые связывают некоторые параметры рабочего процесса (например, зависимость КПД компрессора от его частоты вращения), что позволяет сократить число независимых переменных. Согласно этому подходу, расчет ГТД ведется в нескольких сечениях (на входе в двигатель, на входе в компрессор, за компрессором, за камерой сгорания, за турбиной и на срезе сопла), показанных на рис. 6 и обозначенных соответственно индексами  $H, B, K, \Gamma, T, C$ . Параметры на входе в двигатель определяются скоростью и высотой полета. На первом этапе расчета для каждого сечения последовательно определяются давление и температура газа. При этом должны быть заданы степень повышения давления в компрессоре  $\pi_k$  и температура газа в камере сгорания  $T_\Gamma$ . На данном этапе расчета определяются работа компрессора  $L_k$  и турбины  $L_T$ , расход воздуха, удельная тяга и удельный расход топлива  $C_{уд}$ , который необходим для создания заданной тяги  $R$ . На втором этапе расчета для каждого сечения определяются площади проточной части по формуле  $F_i = G_{\Gamma i} \frac{\sqrt{T_i}}{P_i} \frac{1}{m_{кр}(k_\Gamma, R_\Gamma) q(\lambda_i, k_\Gamma)}$ , где  $i \in \{B, K, \Gamma, T, C\}$ ;  $T_i, P_i$  — температура газа

Таблица 1  
Сравнение различных методов построения приближенных моделей

Модель	Описание построенной модели	Среднеквадратичная погрешность (4)	
		Обучающая выборка	Тестовая выборка
Многослойный персептрон	Два скрытых слоя (11 и 4 нейрона) нейронов с логистической функцией активации, нейрон выходного слоя с линейной функцией активации	2.645	2.960
Метод группового учета аргументов	В качестве опорных функций отобраны полиномы второй и третьей степени и гауссианы	1.016	1.070
Сеть каскадной корреляции	16 нейронов в скрытом слое с гауссовской функцией активации	0.284	0.492
Предложенный метод	26 нейронов с радиальной функцией активации в скрытом слое	0.021	0.064

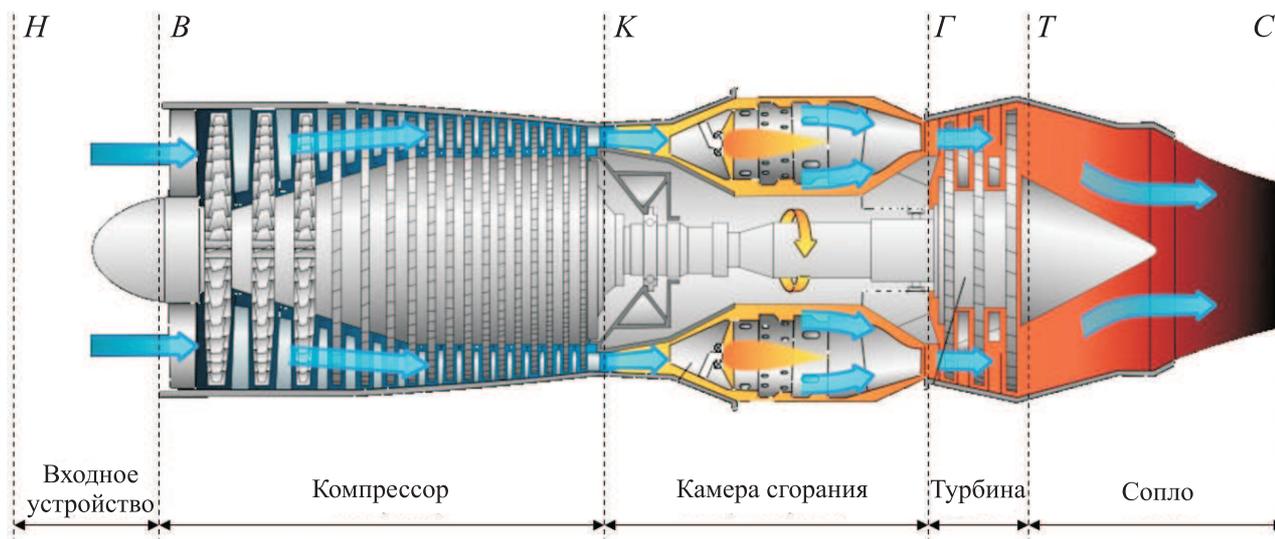


Рис. 6. Модель одноконтурного ГТД

и давление в сечении  $i$ ;  $G_{Gi}$  — расход газа через данное сечение;  $m_{кр}(k, R) = \sqrt{k \left( \frac{2}{k+1} \right)^{(k+1)/(k-1)} \frac{1}{R}}$ ,

$q(\lambda, k) = \lambda \left( 1 - \frac{k-1}{k+1} \lambda^2 \right)^{1/(k-1)} \left( \frac{k+1}{2} \right)^{1/(k-1)}$  — газодинамические функции;  $k_{\Gamma}$  и  $R_{\Gamma}$  — показатель адиабаты и газовая постоянная для газа, протекающего через данное сечение;  $\lambda_i$  — приведенная скорость течения газа в сечении [22]. По известным площадям сечений, а также по заданному внешнему диаметру двигателя  $D$  определяются внутренние и наружные диаметры турбины и компрессора, высоты рабочих лопаток и т.д. Рассмотренный метод расчета подробно описан в работе [5]. Ограничением при определении геометрии двигателя является высота лопатки последней ступени компрессора  $h_z$ , поскольку чрезмерно малая высота приводит к аэродинамическим потерям. В настоящей статье полагается, что должно выполняться условие  $h_z > 15$  мм.

Далее, на основе определенных ранее значений  $L_K$ ,  $L_T$  и максимально возможного значения работы одной ступени определяется число ступеней компрессора  $z_K$  и турбины  $z_T$ , а также частота вращения ротора  $n$ . Данные о частоте вращения и геометрии проточной части позволяют определить напряжения растяжения  $\sigma_p$  в лопатке рабочего колеса последней ступени турбины, которые не должны превосходить

250 МПа [22].

В заключение на основании определенных геометрических параметров компрессора и турбины рассчитывается вес двигателя при помощи эмпирической формулы, полученной в соответствии с рекомендациями [23]:  $W = 4(0.12\rho_{\text{титан}}V_K + 0.25\rho_{\text{сталь}}V_T)$  кг, где  $\rho_{\text{титан}}$ ,  $\rho_{\text{сталь}}$  — плотности титанового сплава и жаропрочной стали и  $V_K$ ,  $V_T$  — геометрические объемы компрессора и турбины.

Таким образом, в соответствии с рассматриваемой математической моделью рабочий процесс одноконтурного ГТД при заданных тяге  $R$  и внешнем диаметре двигателя  $D$  полностью определяется шестью независимыми параметрами:  $\pi_K$  — степень сжатия в компрессоре,  $T_G$  — температура газа (температура в камере сгорания),  $\lambda_B$ ,  $\lambda_K$ ,  $\lambda_G$ ,  $\lambda_T$  — приведенные скорости течения газа за входным устройством, компрессором, камерой сгорания и турбиной соответственно. Ограничениями при выборе допустимого сочетания независимых параметров являются  $h_z$  — высота лопатки последней ступени компрессора и  $\sigma_p$  — напряжения растяжения в лопатке рабочего колеса последней ступени турбины. Зависимости  $C_{уд}$  (кг/Н·ч),  $W$  (кг) и  $\sigma_p$  (кг/мм<sup>2</sup>) от  $\pi_K$  и  $T_G$  (°K) показаны на рис. 7.

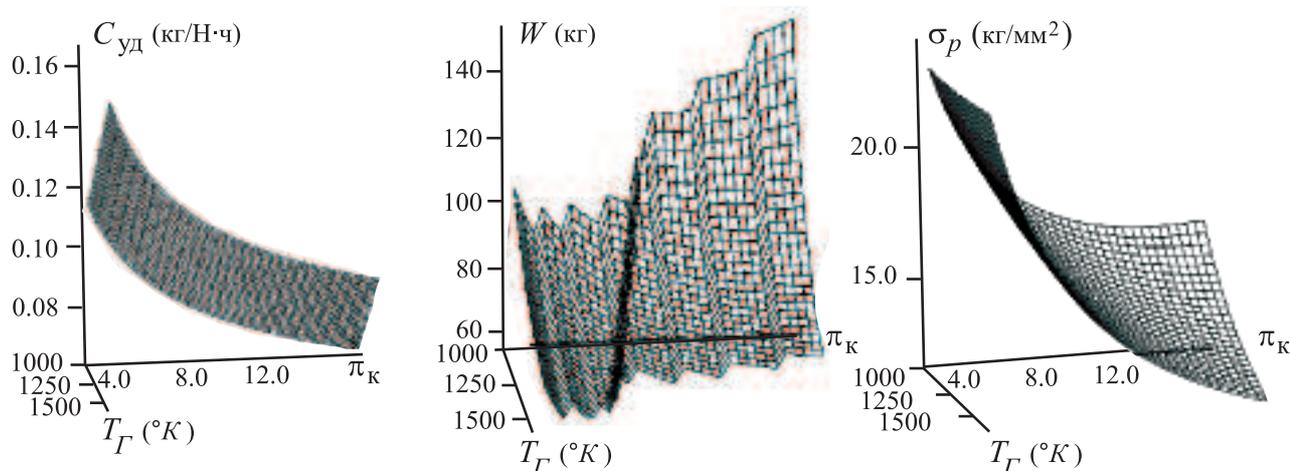


Рис. 7. Влияние независимых переменных на целевые функции и ограничения

**7. Полученные результаты.** Рассмотрим задачу поиска Парето-оптимального множества вариантов конструкции одноконтурного авиационного ГТД с осевым компрессором с тягой на взлетном режиме  $R = 8000$  Н. В качестве целевых переменных, которые необходимо минимизировать, определим удельный расход топлива на взлетном режиме  $C_{уд}$  и вес двигателя  $W$ . Зададим интервалы изменения для независимых переменных: степень сжатия в компрессоре  $\pi_K = 4, \dots, 20$ , температура газа  $T_G = 1300, \dots, 1800$  К, приведенные скорости течения газа  $\lambda_B = 0.6, \dots, 0.7$ ,  $\lambda_K = 0.25, \dots, 0.35$ ,  $\lambda_G = 0.15, \dots, 0.25$ ,  $\lambda_T = 0.4, \dots, 0.65$  и ограничения: высота лопатки последней ступени компрессора  $h_z > 15$  мм и напряжения растяжения в лопатке последней ступени турбины  $\sigma_p > 25$  кг/мм<sup>2</sup>. Расчетная модель двигателя построена описанным выше способом. Зададим  $\varepsilon = 0.005$  в условии окончания вычислений (2).

Процесс решения сформулированной задачи в соответствии с алгоритмом, показанным на рис. 1, представлен в табл. 2. На первом шаге была сгенерирована обучающая выборка из 45 векторов решений  $x = (\pi_K, T_G, \lambda_B, \lambda_K, \lambda_G, \lambda_T)$  в соответствии с центральным композитным планом эксперимента с центрами на гранях (CCF — Central Composite design with Face centered). Из этих 45 решений 9 удовлетворяли ограничениям и 7 являлись недоминируемыми. На основании данной выборки были построены приближенные модели для целевых переменных и ограничений на основе нейронных сетей радиального базиса в соответствии с методом, описанным выше. В табл. 2 для каждой модели приведены количество нейронов в скрытом слое  $N_h$  и приспособленность, вычисленная по формуле (4).

На основании полученных моделей с помощью алгоритма NSGA-II (размер популяции — 100 особей, 500 поколений обучения) было найдено множество из 100 Парето-оптимальных решений, суммарная относительная погрешность (2) при этом составила  $e = 0.0078$ . После проверки данных решений на точной модели, они были добавлены к обучающей выборке, размер которой теперь составил 145 векторов (из них удовлетворяли ограничениям — 41, принадлежали множеству Парето-оптимальных — 14), и весь цикл вычислений был повторен заново (итерация 2). Всего было выполнено 3 итерации, для чего потребовалось 345 вызовов функций (1). Суммарная относительная погрешность моделей, построенных на второй итерации, составила  $e = 0.0069$ , на третьей итерации —  $e = 0.0043$ . Некоторые из найденных Парето-оптимальных параметров рабочего процесса ГТД представлены в табл. 3.

Таблица 2

## Процесс поиска Парето-оптимального множества решений

		1 итерация	2 итерация	3 итерация	Результат
Количество решений в обучающей выборке		45	145	245	345
Количество решений, удовлетворяющих ограничениям		9	41	105	205
Размер Парето-оптимального множества		7	14	22	36
Модель $C_{уд}$	$N_h$	37	39	35	—
	$e_m$	0.00011	0.0001	0.00008	—
Модель $W$	$N_h$	43	41	38	—
	$e_m$	0.17004	0.33003	0.35864	—
Модель $\sigma_p$	$N_h$	38	40	37	—
	$e_m$	0.02941	0.01919	0.02276	—
Модель $h_z$	$N_h$	40	36	37	—
	$e_m$	0.00005	0.00005	0.00004	—
Суммарная относительная погрешность моделей		$e$	0.0078	0.0069	0.0043

Таблица 3

Варианты параметров рабочего процесса одноконтурного ГТД с тягой  $R = 8000$  Н

$C_{уд}$ (кг/Н·ч)	$W$ (кг)	$\sigma_p$ (кг/мм <sup>2</sup> )	$h_z$ (мм)	$\pi_k$	$T_\Gamma$ (°K)	$\lambda_B$	$\lambda_K$	$\lambda_\Gamma$	$\lambda_T$
0.087	141.7	13.0	15	13.2	1311	0.687	0.253	0.25	0.650
0.091	132.3	17.6	15	11.1	1309	0.700	0.289	0.25	0.467
0.093	107.5	17.8	15	10.5	1312	0.700	0.303	0.25	0.470
0.097	97.6	17.6	15	8.9	1312	0.700	0.350	0.25	0.515
0.102	94.5	16.8	15	8.7	1363	0.700	0.350	0.25	0.546
0.111	87.8	17.1	15	8.9	1476	0.700	0.350	0.25	0.515
0.117	84.9	17.2	15	8.8	1552	0.700	0.350	0.25	0.507
0.127	83.1	17.8	15	8.3	1657	0.700	0.350	0.25	0.499
0.132	82.7	17.3	16	8.1	1704	0.700	0.350	0.25	0.528
0.139	80.3	21.9	19	6.7	1696	0.700	0.350	0.25	0.449
0.150	79.3	24.4	24	5.2	1697	0.700	0.350	0.25	0.471

Результаты всех итераций представлены на рис. 8 (в скобках указано количество точек в обучающей выборке/количество точек, принадлежащих Парето-оптимальному множеству решений). На рис. 8 также показано Парето-оптимальное множество (фронт Парето), полученное методом NSGA-II (100 особей в популяции, 500 поколений) на основе точной модели. Для нахождения данного множества потребовалось 50 000 обращений к функциям (1). На рис. 9 представлено сравнение трех Парето-оптимальных множеств решений: полученного на основе предложенной здесь приближенной модели (345 вызовов точной модели) и полученных на основе точной модели за 500 обращений (100 особей в популяции, 5 поколений) и за 50000 обращений (100 особей в популяции, 500 поколений).

Полученные результаты свидетельствуют, что предложенный метод построения приближенных моде-

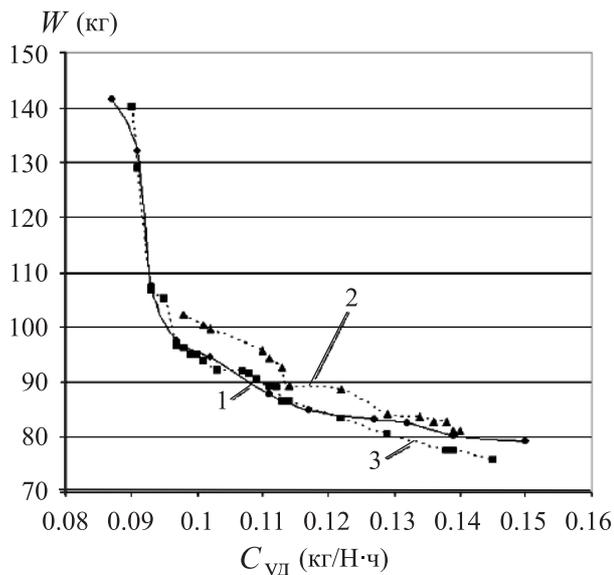
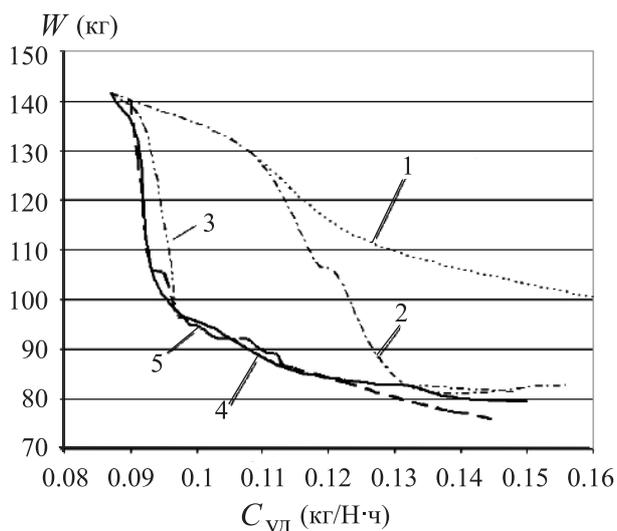


Рис. 8. Эволюция Парето-оптимальных наборов решений в процессе вычислений:

- 1) стартовая обучающая выборка (46/7),
- 2) 1-я итерация (145/14), 3) 2-я итерация (245/22),
- 4) 3-ья итерация (346/36), 5) решение на основе точной модели (50000/100)

Рис. 9. Сравнение трех Парето-оптимальных множеств решений: 1) приближенная модель,

- 2) точная модель — 5 поколений, 3) точная модель — 500 поколений

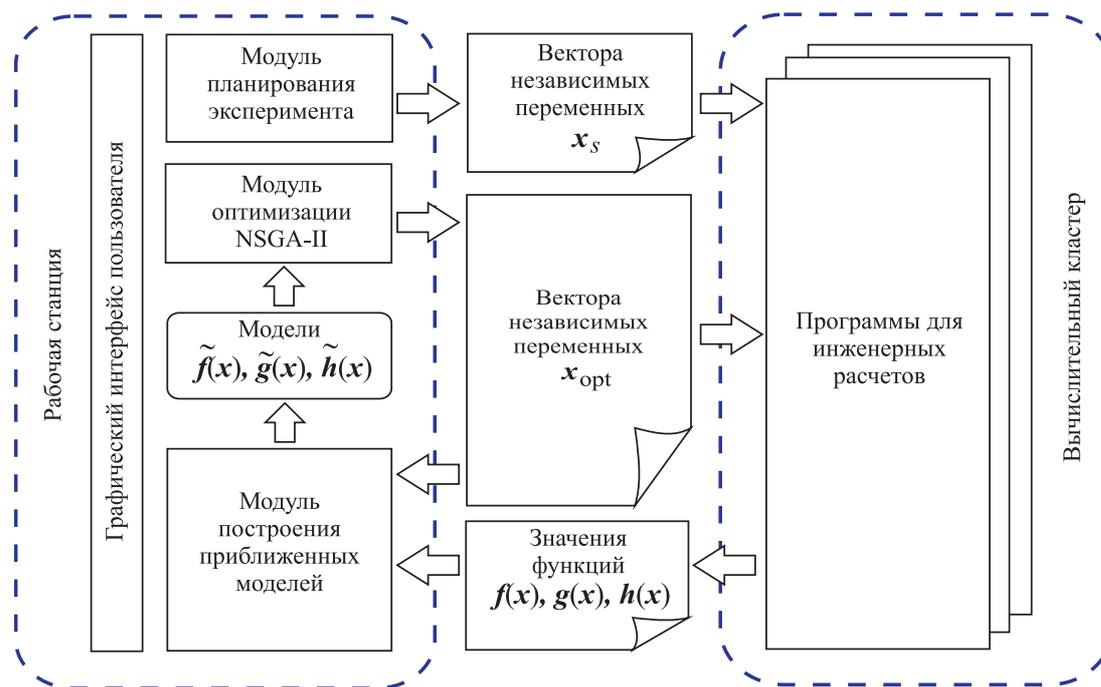


Рис. 10. Программная реализация предложенного метода

лей позволяет сократить затраты машинного времени на расчеты при многокритериальной оптимизации с ограничениями более чем в 100 раз.

**8. Программная реализация предложенного метода.** Программное обеспечение, реализующее описанный метод, написано на языке Python 2.6 с помощью библиотек numpy и scipy. Программа состоит из четырех модулей (рис. 10), реализующих планирование эксперимента, многокритериальную оптимизацию по методу NSGA-II, построение приближенных моделей на базе RBF-сетей, как это было описано выше, и графический интерфейс пользователя. Высокая скорость работы предложенных методов, а так-

же скорость вычислений при помощи упомянутых математических библиотек языка Python позволили разместить все компоненты системы на одной рабочей станции без потери производительности. Обмен с внешними программными системами, в которых осуществляется вычисление точных моделей, ведется через файлы обмена. Данные системы могут быть применены в параллельной среде на вычислительном кластере.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Зеленков Ю.А.* Использование компьютеров в машиностроении. Опыт НПО “Сатурн” // Суперкомпьютерные технологии в науке, образовании и промышленности / Под ред. В.А. Садовниченко, И.Г. Савина, Вл.В. Воеводина. М.: Изд-во Моск. ун-та, 2009.
2. *Васильев Ф.П.* Методы оптимизации. М.: Факториал Пресс, 2002.
3. *Liu G.P., Yang J.B., Whidborne J.F.* Multiobjective optimization and control. Baldock: Research Studies Press Ltd., 2003.
4. *Deb K.* Multi-objective genetic algorithms: problem difficulties and construction of test problems // *Evolutionary Computation*. **7**, N 3. 205–230.
5. Теория и расчет воздушно-реактивных двигателей / Под ред. С.М. Шляхтенко. М.: Машиностроение, 1987.
6. *Зайченко Ю.П.* Исследование операций. Киев: Слово, 2003.
7. *Zitzler E., Thiele L.* Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach // *IEEE Trans. on Evolutionary Computation*, **3**, N 4. 257–271.
8. *Fonseca C.M., Fleming P.J.* Multiobjective optimization and multiple constraint handling with evolutionary algorithms II: application example // *IEEE Trans. on System, Man, and Cybernetics. Part A: System and Humans*. 1998. **28**, N 1. 38–47.
9. *Mosetti C.G., Pediroda V., Poloni C.* Application of evolutive algorithms and statistical analysis in the numerical optimisation of an axial compressor // *The 9th International Symposium on Transport Phenomena and Dynamics of Rotating Machinery*. Honolulu, Hawaii. February 10–14. 2002.
10. *Deb K., Agrawal S., Pratap A., Meyarivan T.* A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II // *In Proc. of the Parallel Problem Solving from Nature VI*. 2000. 849–858.
11. *Egorov I.N., Kretinin G.V., Leshchenko I.A., Kuptcov S.V.* Multi-objective robust optimization using IOSO technology. Part I: Main features // *In Proc. of International Congress on Evolutionary Methods of Design, Optimization and Control with Applications to Industrial Problems EUROGEN 2003*.
12. *Иващенко А.Г.* Индуктивный метод самоорганизации моделей сложных систем. Киев: Наукова Думка, 1981.
13. *Зеленков Ю.А.* Аппроксимация функций на основе радиальных нейронных сетей, генерируемых при помощи эволюционных алгоритмов // *Вестник Рыбинской государственной авиационной технологической академии им. П.А. Соловьева*. Рыбинск, 2004. № 1–2 (4–6). 87–93.
14. *Yao X., Liu Y.* A new evolutionary system for evolving artificial neural networks // *IEEE Trans. Neural Networks*. May 1997. **8**. 694–713.
15. *Зеленков Ю.А., Серебряков П.Г., Шаров В.Г.* Генерация нейронных сетей на основе генетических алгоритмов для задачи классификации образов // *Моделирование и обработка информации в технических системах. Материалы Всероссийской научно-технической конференции*. Рыбинск. РГТА. 2004. 86–89.
16. *Topchy A., Lebedko O., Miagkikh V., Kasabov N.* Adaptive training of radial basis function networks based on co-operative evolution and evolutionary programming // *Progress in Connectionist-Based Information Systems*. N. Kasabov et al (Eds.). Berlin: Springer Verlag, 1998. 253–258.
17. *Fogel D.B.* Evolutionary computation: toward a new philosophy of machine intelligence. New York: IEEE Press, 1995.
18. *Yao X.* Evolving artificial neural networks // *Proc. of the IEEE*. September 1999. **87**, N 9. 1423–1447.
19. *Moody J., Darken C.J.* Fast learning in networks of locally tuned processing units // *Neural Computation*. 1989. **1**. 181–194.
20. *Осовский С.* Нейронные сети для обработки информации. М.: Финансы и Статистика, 2002.
21. *Холщевников К.В., Емин О.Н., Митрохин В.Т.* Теория и расчет авиационных лопаточных машин. М.: Машиностроение, 1986.
22. Конструкция и проектирование авиационных газотурбинных двигателей / Под ред. Д.В. Хронина. М.: Машиностроение, 1989.
23. *Цховербов М.М.* “Модульное” моделирование весовых характеристик ТРДДФ // *ЦИАМ 2001–2005. Основные результаты научно-технической деятельности*. Том I. М.: ЦИАМ, 2005. 64–68.

Поступила в редакцию  
29.08.2010