

**УДК 004.94,519-68**

**Применение генетических алгоритмов в нейро-нечетких  
моделях**

**А.С.Патушин**

**Московский физико-технический институт**

## 1 Введение

В условиях, когда основное количество промышленных предприятий находится в частной собственности и ориентируется на изменчивые нужды рынка и потребительский спрос, моделирование промышленных объектов, в том числе моделирование и прогнозирование потребления электроэнергии становится сложной задачей. Возрастающее энергопотребление, задержки расширения существующих сетей передачи электроэнергии в сочетании с физическим и моральным износом теплоэнергетического оборудования и сетей приводят к тому, что энергосистемы в настоящее время эксплуатируются на пределе своих возможностей. Поэтому прогнозирование параметров электропотребления важно не только для предприятия, но и для энергоснабжающей организации, для эффективного управления которой необходимы графики нагрузки по предприятиям, отдельным узлам энергосистемы и даже отдельным участкам предприятия. Формирование универсального аппарата моделирования и прогнозирования электропотребления и мощности для различных иерархических уровней, способного адекватно и надежно моделировать и прогнозировать потребление электроэнергии, позволило бы эффективно управлять спросом и предложением. Для подобных целей наиболее часто используются регрессионные модели, позволяющие определять участие тех или иных факторов в формировании электропотребления. Но зависимости, качественно моделируемые с помощью регрессионных методов, в большинстве случаев линейны, в то время как большинство потребителей характеризуются нелинейностью зависимостей электропотребления от факторов производства, погодных условий и т.д. Большинство промышленных объектов является сложными слабо формализуемыми системами [5,6], все это обуславливает необходимость применения не только регрессионных методов моделирования в условиях неопределенности связи между входными и выходными переменными. Исследования и практический опыт показывают, что наиболее перспективным направлением в области прогнозирования на

сегодняшний день является использование аппарата искусственных нейронных сетей (ИНС), который позволяет более точно моделировать и прогнозировать и при этом не требует повышенных знаний в области математической статистики, поскольку сама структура нейронной сети выбирается на этапе проектирования модели. Также широкое распространение получили нечеткие множества, которые хорошо описывают динамические свойства энергетических объектов. На современном этапе развития сложных электроэнергетических систем становится особенно актуальным максимальное снижение затрат на решение задач, связанных с управлением, оптимизацией и планированием их режимов. Существующая вычислительная техника и новые методы и модели программирования позволяют реализовать довольно сложные, но в то же время эффективные алгоритмы. Повышенное внимание уделяется эволюционным алгоритмам, наиболее известными эволюционными методами являются генетические алгоритмы. Генетические алгоритмы широко используются в нейронных сетях при обучении сети для нахождения оптимальных значений коэффициентов весов. Использование генетических алгоритмов обусловлено их высокой эффективностью и меньшим временем работы по сравнению с классическими алгоритмами, что оказывается ценным в энергетических системах работающих в режиме реального времени или близком к нему.

## **2 Обзор методов нечеткой логики и нейронных сетей**

### **2.1 Основные понятия генетических алгоритмов**

Генетический алгоритм это алгоритм позволяющий, найти решение аналитически не разрешимых задач, используя последовательный подбор и комбинирование исходных параметров с использованием механизмов, похожих на биологические.

Генетические алгоритмы применяются для решения следующих задач:

- Оптимизация функций
- Разнообразные задачи на графах (задача коммивояжера, раскраска, нахождение паросочетаний) [25]
- Настройка и обучение искусственной нейронной сети [13]
- Задачи компоновки [24]
- Составление расписаний [14]
- Игровые стратегии [20]
- Аппроксимация функций [19]
- Искусственная жизнь [13]
- Биоинформатика

Классический генетический алгоритм состоит из следующих шагов

- 1 Инициализация ( выбор исходной популяции хромосом)
- 2 Оценка приспособленности хромосом
- 3 Проверка условия останова алгоритма
- 4 Селекция хромосом
- 5 Применение генетических операторов
- 6 Формирование новой популяции
- 7 Выбор наилучшей хромосомы

Инициализация – формирование исходной популяции, заключается в случайном выборе заданного количества хромосом.

Оценивание приспособленности хромосом в популяции состоит из расчета функции приспособленности для каждой хромосомы. Чем выше значение

функции приспособленности, тем выше качество хромосомы. Эта функция выбирается исходя из конкретной задачи.

Селекция хромосом заключается в выборе хромосом по значениям функции приспособленности, которые будут участвовать в создании потомков для следующей популяции. Существуют различные методы селекции, наиболее популярным является метод рулетки, заключающийся в том, что каждой хромосоме сопоставляется сектор колеса рулетки, пропорциональный функции приспособленности. Поэтому, чем больше значение функции приспособленности, тем больше сектор. Все колесо рулетки соответствует сумме значений функций приспособленности для всех хромосом. Каждой хромосоме  $ch_i$   $i = 1, 2 \dots N$  ( где  $N$  – численность популяции ) соответствует сектор колеса  $v(ch_i)$ , выраженный в процентах

$$v(ch_i) = p_s(ch_i) * 100\%$$

где  $p_s(ch_i) = \frac{F(ch_i)}{\sum_{i=1}^N F(ch_i)}$  - вероятность селекции хромосомы,  $F(ch_i)$  – значение функции приспособленности для хромосоме  $ch_i$ .

Селекция хромосом может быть представлена случайным поворотом колеса рулетки. Каждая особь получает в родительском пуле количество копий установленных выражением

$$e(ch_i) = p_s(ch_i) * N$$

С другой стороны  $e(ch_i) = \frac{F(ch_i)}{\bar{F}}$ , где  $\bar{F}$  – среднее значение функции приспособленности. Согласно этому метод рулетки можно применять, когда среднее значение функции приспособленности положительно. Один из недостатков метода рулетки – быстрое удаление из популяции хромосом с маленькой функцией приспособленности. Есть методы не обладающие данным недостатком – турнирный метод и метод ранговой селекции.

При турнирной селекции все особи популяции разбиваются на подгруппы с последующим выбором в каждой из них наилучшей хромосомы. Различаются два способа такого выбора : детерминированный ( вероятность равна 1 ) и случайный ( вероятность выбора меньше 1). Подгруппы могут иметь произвольный размер, но чаще всего используются подгруппы с 2-3 особями.

При ранговой селекции особи популяции ранжируются по значениям функции приспособленности. Количество копий введенных в родительскую популяцию рассчитывается по априорно заданной функции, в зависимости от ранга.

Оператор скрещивания. На первом этапе скрещивания выбираются пары хромосом из родительской популяции. Эта временная популяция, состоящая из хромосом отобранных в результате селекции и предназначенных для дальнейших преобразований скрещивания и мутации. Хромосомы случайным образом объединяются в пары, в каждой паре определяется точка скрещивания. В результате скрещивания появляется новая популяция потомков. В классическом генетическом алгоритме операция скрещивания представляет собой точечное скрещивание, также применяются другие виды скрещивания – двухточечное, многоточечное и равномерное. Двухточечное скрещивание отличается от одноточечного тем, что потомки наследуют фрагменты родительских хромосом определяемые двумя случайно выбранными точками скрещивания. Многоточечное скрещивание представляет собой обобщение предыдущих операций и характеризуется большим количеством точек скрещивания. Равномерное скрещивание выполняется в соответствии со случайно выбранным эталоном, который указывает какие гены должны наследоваться от первого родителя а какие от второго.

## 2.2 Обзор моделей генетических алгоритмов

### 1. Genitor

В данной модели используется стратегия отбора, при которой каждая пара особей создает только одного ребенка. Этот ребенок заменяет на родителей, а одну из худших особей в популяции. В итоге в популяции на каждом шаге обновляется только одна особь. Исследование показали, что данная модель имеет лучшую сходимость, чем классический генетический алгоритм. В работе [18] проводится сравнение harmony search алгоритма и genitor модели. В результате сравнение этих модификаций генетического алгоритма получаем, что использовании genitor модели число дает меньшее число шагов генетического алгоритма, в сравнении с harmony search.

### 2. СНС

СНС – это Cross generational elitist selection, Heterogenous recombination, Cataclysmic mutation.

Для нового поколения выбираются N различных лучших особей среди родителей и детей. В скрещивании участвуют пары особей расстояние Хемминга для которых больше порогового значения. При скрещивании используется так называемый HUX-оператор (Half Uniform Crossover), разновидность однородного кроссовера – каждому потомку переходит ровно половина битов каждого родителя. Данный алгоритм быстро сходится, т.к в нем нет мутаций. Чтобы этого избежать используют сильную мутацию – самая приспособленная особь остается без изменений, а все остальные подвергаются сильной мутации, и алгоритм запускается снова. В работе [15] проводится сравнение обычного генетического алгоритма и улучшенного с использованием СНС генетического алгоритма. Производительность СНС генетического алгоритма значительно выше, чем обычного генетического алгоритма.

### 3. Hybrid algorithm

Поиск оптимального решения в генетических алгоритмах может занимать большой период времени, чтобы этого избежать генетические алгоритмы используют вместе с классическими алгоритмами оптимизации. Сначала генетический алгоритм сходится в маленькую область, а в этой области используем какой-либо классический алгоритм. В работе [8] проводится сравнение гибридного генетического алгоритма и обычного генетического алгоритма. По результатам тестирования алгоритмов, гибридный генетический алгоритм показал производительность выше, чем не модифицированный генетический алгоритм.

#### 4. Island Models

В данной модели исходная популяция разбивается на несколько подмножеств, которые будут развиваться отдельно друг от друга. Изредка будет происходить обмен хорошими особями между подмножествами. В данной модели очень важно правильно задать частоту миграции. В работе [22] рассматривается улучшение island model для генетического алгоритма.

### 2.3 Применение генетических алгоритмов в нейронных сетях

Объединение генетических алгоритмов и нейронных сетей известно в литературе под аббревиатурой COGANN (Combinations of Genetic Algorithms and Neural Networks) [4]. Это объединение может быть вспомогательным (supportive) либо равноправным (collaborative). Вспомогательное объединение двух методов означает, что они применяются последовательно один за другим, причем один из них служит для подготовки данных, используемых при реализации второго метода. При равноправном объединении оба метода применяются одновременно.

Вид объединения	Характеристика объединения	Примеры использования
-----------------	----------------------------	-----------------------



	Генетические алгоритмы и нейронные сети независимо применяются для решения одной и той же задачи	Однонаправленные нейронные сети, сети Кохонена с самоорганизацией и генетические алгоритмы в задачах классификации
Вспомогательное	Нейронные сети для обеспечения генетических алгоритмов	Формирование исходной популяции для генетического алгоритма
	Генетические алгоритмы для обеспечения нейронных сетей	Подбор параметров либо преобразование пространства параметров
Равноправное	Генетические алгоритмы для обучения нейронных сетей	Эволюционное обучение сети (эволюция весов связей)
	Генетические алгоритмы для выбора топологии нейронной сети	Эволюционный подбор топологии сети (эволюция сетевой архитектуры)
	Системы, объединяющие адаптивные стратегии генетических алгоритмов и нейронных сетей	Нейронные сети для решения оптимизационных задач с применением генетического алгоритма для подбора весов сети
		Реализация генетического алгоритма с помощью нейронной сети
		Применение нейронной сети для реализации

		оператора скрещивания в генетическом алгоритме
--	--	--

Генетические алгоритмы и нейронные сети могут независимо применяться для решения одной и той же задачи. Например, описаны независимые применения нейронных сетей, генетических алгоритмов и алгоритма KNN «ближайший сосед» (K - means nearest neighbour) для решения задач классификации.

Нейронная сеть для обеспечения генетических алгоритмов может использоваться для решения задачи трассировки, которая классифицируется как пример вспомогательного объединения нейронных сетей и генетических алгоритмов. В этой системе генетический алгоритм используется в качестве оптимизационной процедуры, предназначенной для нахождения кратчайшего пути. Нейронная сеть применяется при формировании исходной популяции для генетического алгоритма.

Подход, основанный на использовании генетического алгоритма для обеспечения работы нейронной сети применяется для решения проблем [16]

- применение генетического алгоритма для подбора параметров либо преобразования пространства параметров, используемых нейронной сетью для классификации;
- применение генетического алгоритма для подбора правила обучения либо параметров, управляющих обучением нейронной сети;
- применение генетического алгоритма для анализа нейронной сети.

Применение генетических алгоритмов для обучения нейронных сетей, как задача заключается в оптимизации весов нейронной сети, имеющей априори заданную топологию [23].

Генетические алгоритмы для выбора топологии нейронных сетей. Проектирование оптимальной топологии нейронной сети может быть представлено в виде поиска такой архитектуры, которая обеспечивает наилучшее (относительно выбранного критерия) решение конкретной задачи. Такой подход предполагает перебор пространства архитектур, составленного из всех возможных вариантов, и выбор точки этого пространства, наилучшей относительно заданного критерия оптимальности. При этом в генотипе кодируется топология сети [10].

Нейронные сети с генетическими алгоритмами могут решать задачи

Задача наилучшего приближения: если рассматривать систему  $n$  линейных уравнений с  $m$  неизвестными  $Ax = b$  в случае, когда она переопределена ( $n > m$ ), то иногда оказывается естественной задача о нахождении вектора  $x$ , который "удовлетворяет этой системе наилучшим образом", т. е. из всех "не решений" является лучшим.

Задача о рационе: пусть имеется  $n$  различных пищевых продуктов, содержащих  $m$  различных питательных веществ. Обозначим через  $a_{ij}$  содержание (долю)  $j$ -го питательного вещества в  $i$ -ом продукте, через  $b_j$  — суточную потребность организма в  $j$ -ом питательном веществе, через  $c_i$  — стоимость единицы  $i$ -го продукта. Требуется составить суточный рацион питания минимальной стоимости, удовлетворяющий потребность во всех питательных веществах

Транспортная задача: эта задача — классическая задача линейного программирования. К ней сводятся многие оптимизационные задачи. Формулируется она так. На  $m$  складах находится груз, который нужно развезти  $n$  потребителям. Пусть  $a_i$  ( $i = 1, \dots, n$ ) — количество груза на  $i$ -ом складе, а  $b_j$  ( $j = 1, \dots, m$ ) — потребность в грузе  $j$ -го потребителя,  $c_{ij}$  — стоимость перевозки единицы груза с  $i$ -го склада  $j$ -му потребителю. Требуется минимизировать стоимость перевозок.

Задачи о распределении ресурсов: общий смысл таких задач — распределить ограниченный ресурс между потребителями оптимальным образом.

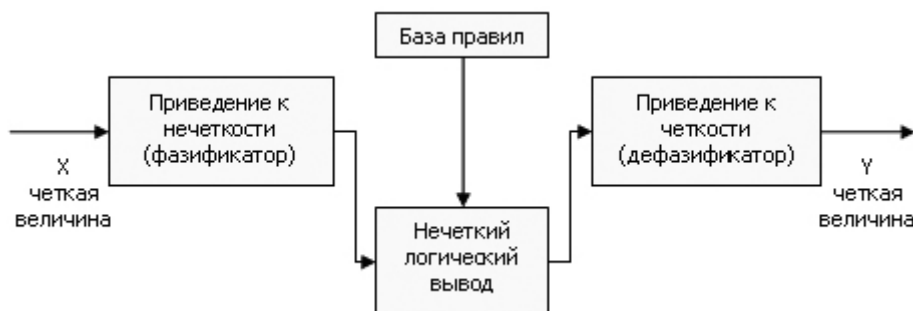
Рассмотрим простейший пример — задачу о режиме работы энергосистемы.

Пусть  $m$  электростанций питают одну нагрузку мощности  $p$ . Обозначим через  $x_j$  активную мощность, генерируемую  $j$ -ой электростанцией.

Техническими условиями определяются возможный минимум  $m_j$  и максимум  $M_j$  вырабатываемой  $j$ -ой электростанцией мощности. Допустим затраты на генерацию мощности  $x$  на  $j$ -ой электростанции равны  $e_j(x)$ . Требуется сгенерировать требуемую мощность  $p$  при минимальных затратах.

## 2.4 Обзор методов нечеткой логики

Часто вместе с нейронными сетями используется модуль нечеткого управления. Классический модуль нечеткого управления состоит из блока фаззификации, базы правил, блока вывода и блока дефаззификации [2].



База правил это множество нечетких правил  $R^k$ , где  $k = 1, \dots, N$  вида

$R^k$ : Если ( $x_1$  это  $A_1^k$  и  $x_2$  это  $A_2^k$  ... и  $x_n$  это  $A_n^k$ ) Тогда ( $y_1$  это  $B_1^k$  и  $y_2$  это  $B_2^k$  ... и  $y_m$  это  $B_m^k$ ), где  $N$ - количество нечетких правил,  $A_i^k, B_i^k$ - нечеткие множества

$x_1, x_2, \dots, x_n$ - входные переменные нечеткой модели

примем  $(x_1, x_2, \dots, x_n)^T = x \in X_1 \times X_2 \times \dots \times X_n$

$y_1, y_2, \dots, y_m$ - выходные переменные лингвистической модели

примем  $(y_1, y_2, \dots, y_m)^T = y \in Y_1 \times Y_2 \times \dots \times Y_m$

Каждое правило состоит из части Если называемого посылкой и части Тогда называемой следствием. Посылка содержит условие, а следствие содержит вывод.

Блок фаззификации: система управления с нечеткой логикой оперирует нечеткими множествами, поэтому конкретное значение  $\bar{x} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)^T$  входного сигнала подлежит фаззификации, в результате которой ему будет сопоставлено нечеткое множество  $A$ . Эта операция осуществляется с помощью вычисления функции принадлежности. Гауссова функция принадлежности имеет вид

$$\mu(x) = \exp\{-(x - b_1/b_2)^2\}$$

Блок выработки решений

Допустим, что на вход блока выработки решения подано нечеткое множество  $A$ . На выходе этого блока также появится соответствующее нечеткое множество. На выходе блока выработки решения в соответствии с обобщенным нечетким правилом получаем  $N$  нечетких множеств  $\bar{B}^k$ . При этом можно задать функцию принадлежности

$$\mu_{B^k}(x) = \sup(\mu_A(x)^T * \mu_{A^k \rightarrow B^k}(x, y))$$

Блок дефаззификации

На выходе блока выработки решения формируется  $N$  нечетких множеств  $B^k$  с функциями принадлежности  $\mu_{B^k}(y)$ . Значение  $\bar{y}$  можно рассчитать следующими способами:

1. Метод дефаззификации по среднему центру

$$\bar{y} = \frac{\sum_{i=1}^N \mu_{B^k}(y^k) y^k}{\sum_{i=1}^N \mu_{B^k}(y^k)}, \text{ где } y^k \text{ — точка в которой } \mu_{B^k} \text{ принимает}$$

максимальное значение.

2. Метод дефаззификации по сумме центров

$$\bar{y} = \frac{\int_Y y \sum_{i=1}^N \mu_{B^k}(y) dy}{\int_Y \sum_{i=1}^N \mu_{B^k}(y) dy}$$

### 3. Метод центра тяжести

$$\bar{y} = \frac{\int_Y y \mu_{B^k}(y) dy}{\int_Y \mu_{B^k}(y) dy}$$

## 2.5 Обзор применения генетических алгоритмов и нейронных сетей в энергетике

В работе [17] рассматривается моделирование спроса на электричество, используя две модели: нейронную сеть и экономический подход со статистическими методами. Первым шагом является определение ключевых переменных модели. Эти переменные – цена 1квт/ч электричества, цена на газ, и потребительский спрос за прошлые годы. В работе показывается прямая зависимость между потреблением в прошлом году и предсказанным потреблением. Также в прямой зависимости находятся цена на газ и текущее потребление, напротив цена на электричество и потребление находятся в косвенной зависимости, подтверждающейся отрицательным наклоном кривой спроса. Выбор данных переменных позволяет получить предсказания, удовлетворяющие заданной точности. Модель с нейронной сетью дает ошибку 5%, а экономическая модель – 8%.

В работе [8] для оценки спроса на электроэнергию используется генетический алгоритм и нейронная сеть, использующие стохастические процедуры. Параметрами модели являются цена на электроэнергию, количество клиентов и потребление за прошлые года. Модель может быть использована в будущем путем оптимизации значений параметров. Генетический алгоритм был настроен для всех его параметров и были найдены оптимальные значения коэффициентов. Нейронная сеть использовалась для прогнозирования значений параметров, а затем и для прогнозирования спроса.

В октябре 2012 вышла работа [12] в которой используется искусственная нейронная сеть для определения потребления обычными жителями на один час вперед. Для этих целей использовался Smart Meter (SM), метеорологические данные, получаемые в реальном времени и нейронная сеть. Система использовалась в одном из районов Афин. SM использовался для определения и записи потребителей. Одновременно Национальная Обсерватория поставляла метеорологические данные. Эти данные поступали в модель для предсказания спроса на электричество на один час вперед. После обучения искусственной нейронной сети на предложенной выборке, она использовалась для предсказания потребления. Результаты показали, что 98% предсказанных значения отличаются от реальных значений на  $\pm 10\%$ .

В статье [26] используется улучшенная нейронная сеть с обратным распространением ошибки для предсказания спроса на электроэнергию в Китае. Matlab7 использовался как средство моделирования. В качестве параметров модели использовались рост ВВП в текущем году и потребление электричества за текущий год для предсказания потребления электричества в следующем году. В работе сравниваются результаты работы предложенной сети и обычной. Улучшенная нейронная сеть дает более точные предсказания и меньшую ошибку.

В работе [5] используются нечеткая нейронная сеть вместе с генетическим алгоритмом для определения стоимости недвижимости. Стоимость недвижимости очень сложно взаимосвязана с множеством факторов. Генетический алгоритм использовался для оптимизирования коэффициентов для нейронной сети. В работе представлены результаты, касающиеся эффективности и времени работы модели, показывающие обоснованность использования нечеткой нейронной сети с генетическим алгоритмом. В работе показывается применимость нейронной сети для решения данной задачи.

В работе [21] проводится моделирование спроса на электроэнергию в Индии с использованием генетического алгоритма. Параметрами модели является население Индии, GDP, импорт и экспорт. В будущем построенная модель может легко быть модифицирована при добавлении новых переменных.

### 3 Нейро-нечеткое моделирование систем

#### 3.1 Постановка задачи

Пусть у нас есть  $N$  входных измеряемых переменных. Они образуют входное множество сигналов  $X$ . Поскольку размерность большая то, разделим входное множество  $X$ , на два произвольных подмножества  $X_1$  и  $X_2$ . Для обоих подмножеств промежуточная модель будет выдавать выходные значения  $z_1$  и  $z_2$ , которые будут являться входами основной модели.

Тогда вместо построения модели с  $N$  входными переменными, основная модель будет с двумя входными переменными. Выходной сигнал основной модели будем обозначать через  $y$ .

$y = f(z_1, z_2)$  где  $z_1$  и  $z_2$  – значения, полученные из промежуточной модели.

В итоге у нас получится множество выходных сигналов основной модели  $y_i$ .

Будем искать модели удовлетворяющие условию

$(\hat{y} - y_i)^2 < \varepsilon$ , где  $\hat{y}$  – реальное значение выходных сигналов.

Если ни одна модель не будет удовлетворять условию, тогда применим следующее приближения

$$\tilde{y} = \sum_i^N b_i y_i$$

Проверим  $\tilde{y}$  на выполнения условия, описанного выше. Рассмотрим всевозможные варианты разбиения  $X$  на  $X_1$  и  $X_2$  и среди всех моделей



получившихся моделей необходимо выбрать модель, которая удовлетворяет условию.

### 3.2 Описание сети ANFIS

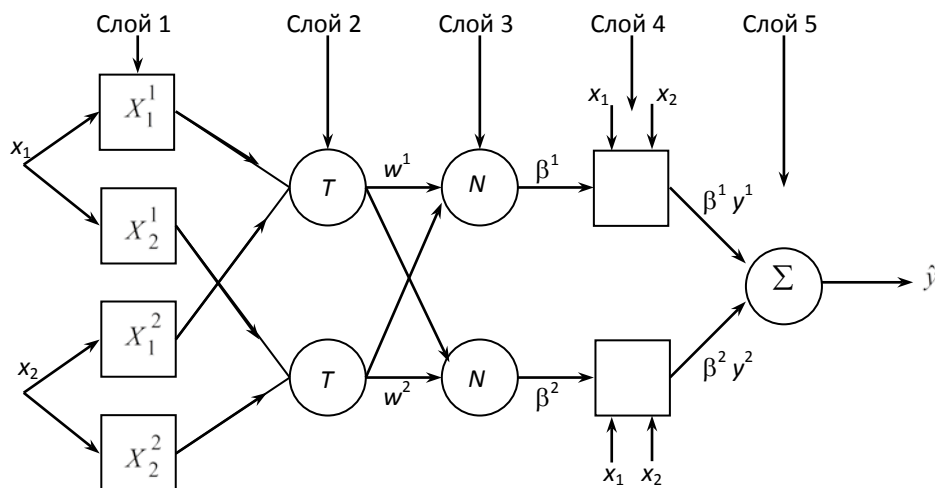
Нейро-нечеткие модели могут быть двух типов совместная нейро-нечеткая модель и параллельная нейро-нечеткая модель. В простейшем случае совместная модель [7,22] понимается как, препроцессор, где механизм обучения искусственной нейронной сети (ANN) определяет правила нечеткого вывода (FIS). Как только параметры FIS определяются, ANN работает в обычном режиме. В параллельной модели нейронная сеть используется, чтобы определить параметры, требующиеся нечеткой системе. Обучение происходит только в нейронной сети, нечеткая система остается неизменной.

Рассмотрим нейро-нечеткие модели со структурой ANFIS (Adaptive Neuro-Fuzzy Inference System)[3], реализующую алгоритм нечеткого вывода Такаги-Сугено. Для простоты рассмотрим нечеткие правила, которые имеют только две входные переменные:

$$\Pi_1 : \text{Если } x_1 \text{ есть } A_{11} \text{ и } x_2 \text{ есть } A_{12}, \text{ То } y_1 = c_{11}x_1 + c_{12}x_2$$

$$\Pi_2 : \text{Если } x_1 \text{ есть } A_{21} \text{ и } x_2 \text{ есть } A_{22}, \text{ То } y_1 = c_{21}x_1 + c_{22}x_2$$

Тогда структура нейро-нечеткой модели будет иметь вид



Слой 1. Выходы слоя отображают значения функций принадлежности  $\mu_{A_{ij}}(x_j)$ , при заданных значениях входных переменных.

Слой 2. Элементы слоя выполняют агрегирование степеней истинности предпосылок каждого правила, в соответствии с операцией Т-нормы, в качестве которой может использоваться операция минимума:

$$\alpha_1 = \min(A_{11}(x_1), A_{12}(x_2))$$

$$\alpha_2 = \min(A_{21}(x_1), A_{22}(x_2))$$

Слой 3. Используется для нахождения  $\beta_i$  – нормализованные значения для  $\alpha_i$ :

$$\beta_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2}, \quad \beta_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2}$$

Слой 4. Узлы данного слоя определяются линейными (для модели типа Сугено) функциями принадлежности выходных переменных:

$$\beta'_i = \beta_i y_i, \text{ где } \beta'_i - \text{выходы 4 слоя.}$$

Слой 5. Единственный узел данного слоя является фиксированным узлом, в котором вычисляется полное выходное значение адаптивной сети как сумма всех входных сигналов  $y' = \frac{\alpha_1 y_1 + \alpha_2 y_2}{\alpha_1 + \alpha_2} = \beta_1 y_1 + \beta_2 y_2$

Если задано четкое множество примеров обучающей выборки, то параметры сети  $a_{ij}, b_{ij}$ , определяющие форму функции принадлежности могут быть настроены с использованием алгоритмов обучения, например на основе градиентных методов.

В качестве функций принадлежности, задающих нечеткие множества, можно использовать

$$\text{сигмоидную } \mu(x) = (1 + \exp(b_1(x - b_2)))^{-1}$$

гауссовую

$$\mu(x) = \exp\{-(x - b_1/b_2)^2\}$$

колокольную

$$\mu(x) = (1 + ((x - b_1)/b_2)^{2b_3})^{-1}$$

и другие зависимости [11].

### 3.3 Методика обучения нейронной сети

Параметры подвергаемы настройке разделяются на два типа параметры  $c_{ij}$  и параметры нелинейных функций принадлежности  $a_{ij}$  и  $b_{ij}$ . Сначала рассчитываются параметры  $c_{ij}$  при фиксированных значениях параметров нелинейных функций принадлежности.

Выходная функция имеет вид:

$$y' = \sum_{i=1}^N \beta_i \sum_{j=1}^N c_{ij} x_j$$

где  $\beta_i = \alpha_i / \sum_{j=1}^N \alpha_j$

При  $K$  обучающих примерах  $(x_1^k, x_2^k, \dots, x_m^k, y^k)$  где  $k = 1, \dots, K$ , и замене значений выходных переменных  $y'^k$  эталонными значениями  $y^k$ , получим систему из  $K$  линейных уравнений. В матричной виде эта система выглядит

$$\begin{pmatrix} w_1^{(1)} & w_1^{(1)} x_1^{(1)} & w_1^{(1)} x_2^{(1)} & \dots & w_1^{(1)} x_m^{(1)} & \dots & w_n^{(1)} & w_n^{(1)} x_1^{(1)} & \dots & w_n^{(1)} x_m^{(1)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ w_1^{(k)} & w_1^{(k)} x_1^{(k)} & w_1^{(k)} x_2^{(k)} & \dots & w_1^{(k)} x_m^{(k)} & \dots & w_n^{(k)} & w_n^{(k)} x_1^{(k)} & \dots & w_n^{(k)} x_m^{(k)} \end{pmatrix}^* = \begin{pmatrix} c_{10} \\ \dots \\ c_{1m} \\ \dots \\ c_{n0} \\ \dots \\ c_{nm} \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \dots \\ y^{(k)} \end{pmatrix}$$

$$w_i = \frac{\prod_{j=1}^m \mu_{A_{ij}}(x_j)}{\sum_{r=1}^n \prod_{j=1}^m \mu_{A_{rj}}(x_j)}$$

$w_i^{(k)}$  – агрегированная степень истинности предпосылок по  $i$  правилу при входном векторе  $(x_1^{(k)}, \dots, x_m^{(k)})$

Где  $m$ - число входных переменных,  $n$ - число правил

так  $W * c = y$ . Размерность матрицы  $W$  равна  $K \times (m + 1)n$ , причем обычно количество строк  $K$  намного больше количество столбцов  $(m + 1)n$ . Решение этой системы уравнений можно провести за один шаг при помощи псевдоинверсии матрицы  $W$

$$c = W^+ y = (W^T W)^{-1} W^T y$$

Затем после определения коэффициентов  $c_{ij}$  их фиксируют и рассчитывают фактические выходные сигналы сети для всех примеров, для чего используется линейная зависимость

$$y' = W * c$$

Затем определяется вектор ошибки

$$e = y' - y$$

После этого по алгоритму обратного распространения ошибки уточняются коэффициенты функции принадлежности  $a_{ij}$  и  $b_{ij}$ .

$$a_{ij}^k(t + 1) = a_{ij}^k(t) - \eta \frac{\partial E^k(t)}{\partial a_{ij}^k}$$

$$b_{ij}^k(t + 1) = b_{ij}^k(t) - \eta \frac{\partial E^k(t)}{\partial b_{ij}^k}$$

где  $t$  – номер очередной итерации.

После уточнения параметров функции принадлежности весь процесс повторяется вновь и так продолжается пока параметры не стабилизируются. Данный алгоритм является гибридным алгоритмом, т.к процесс обучения разделяется на два этапа, на каждом этапе уточняется только часть параметров сети.

### 3.4 Описание генетического алгоритма

Генетический алгоритм основывается на биологических принципах эволюции, классический генетический алгоритм содержит операторы формирования начальной популяции, операторы мутации, скрещивания и селекции. В качестве входных переменных использовались значения  $\hat{y}$  полученные из нейронечеткой модели. Функция приспособленности, отвечающая за селекцию хромосом имеет вид

$$fitness = \sum_{i=1}^{30} (y_i - \sum_{j=1}^2 x_j \hat{y}_{ij})$$

Где  $y_i$ - реальные значения температуры смотки.

В качестве метода селекции использовался метод “рулетки”, заключающийся в том, что каждой хромосоме сопоставляется сектор колеса рулетки, пропорциональный функции приспособленности. Поэтому, чем больше значение функции приспособленности, тем больше сектор. Все колесо рулетки соответствует сумме значений функций приспособленности для всех хромосом.

В качестве мутации использовался Гауссовский метод мутации, заключающийся в том, что к  $x$  добавлялось случайное значение гауссовского распределения с центром в нуле. Среднеквадратичное отклонение контролируется двумя параметрами. Параметр масштаба отвечает за среднеквадратичное отклонение. Параметр сокращения отвечает за уменьшение среднеквадратичного отклонения с ростом числа поколений. Оба параметра были равны единице.

Для кроссовера использовался оператор, который сначала создает случайным образом вектор, показывающий какие гены будут взяты от одного родителя, какие от другого.

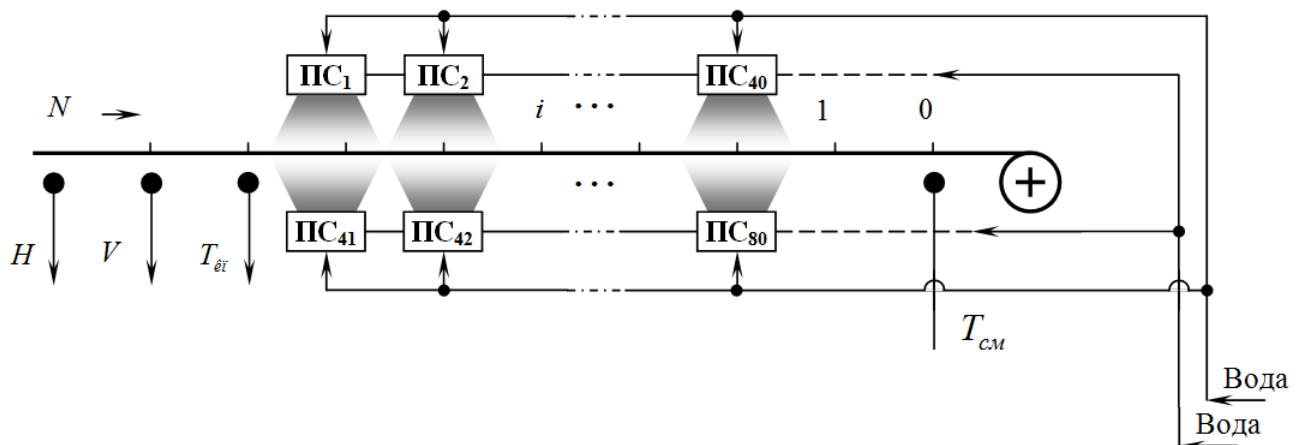
Условием остановки генетического алгоритма – число поколений равно 100 или  $fitness < 10^{-3}$ .

## 4 Моделирование душирующей установки

### 4.1 Описание душирующей установки

В металлургическом производстве на непрерывных широкополосных станах горячей прокатки, основной из основных установок является установка ускоренного охлаждения, называемая душирующей установкой. Физические процессы происходящие в ней довольно сложны, поэтому строгое математическое описание душирующей установки тоже сложно.

Душирующая установка изображена на рисунке



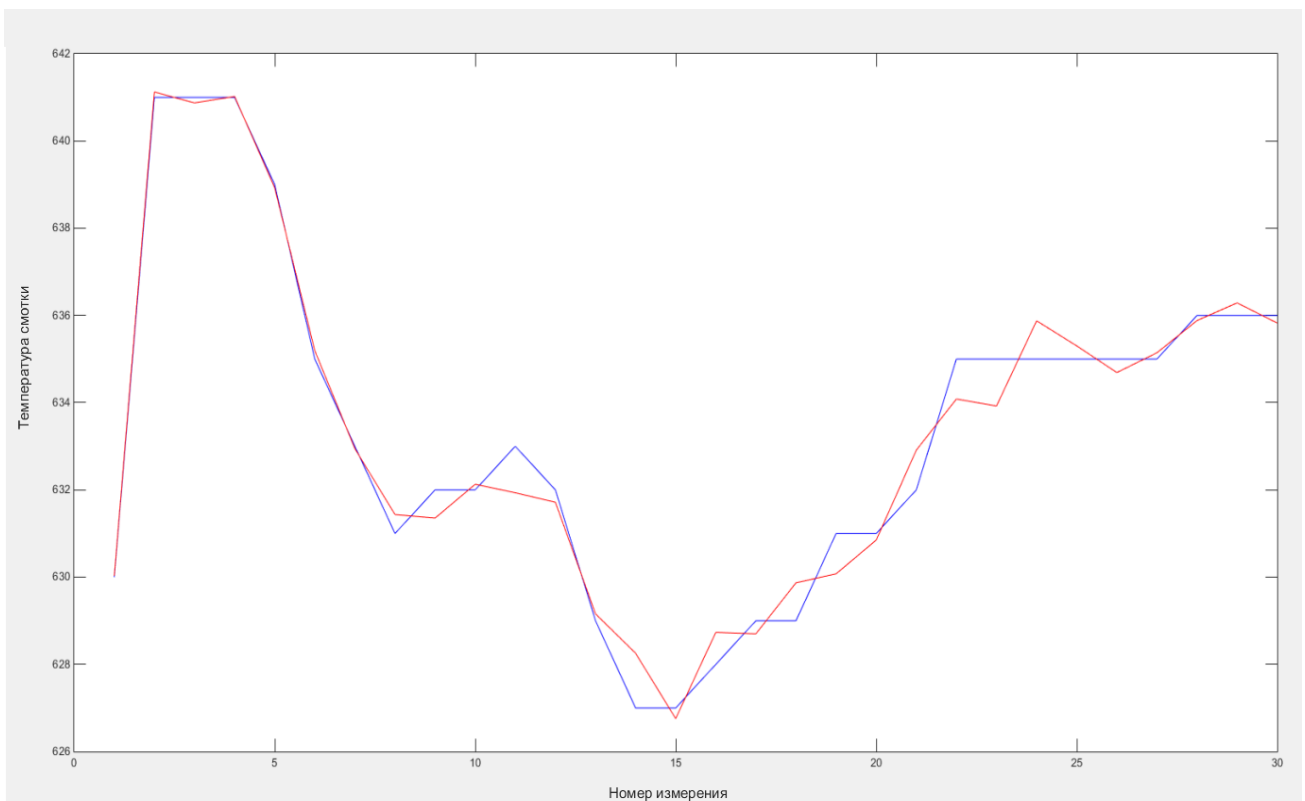
В зависимости от толщины  $H$  и скорости  $V$  полосы, температуры конца прокатки  $T_{кп}$  включается определенное число  $N_0 = \overline{1,80}$  полусекций  $PS_1, \dots, PS_{80}$  охлаждающих горячую полосу до температуры смотки  $T_{см}$ . Затем охлажденная полоса сматывается в рулон. Математическая модель душирующей установки будет представлять собой зависимость температуры смотки  $T_{см}$  от входных переменных количества включенных полусекций  $N_0$ ,

толщины  $H$  и скорости  $V$  полосы, температуры конца прокатки  $T_{\text{кп}}$ . При моделировании рассмотрим две модели. В качестве первой модели будем использовать сеть ANFIS с шестью входными переменными  $N_0(i), N_0(i - 1), V(i), V(i - 1), H(i), T_{\text{кп}}(i)$ . Для второй модели используем метод двухступенчатой идентификации. Соответственно будет две подмодели, каждая из которых включает в себя три входные переменные, объединенные посредством формулы

$\tilde{y} = \sum_i^2 b_i y_i$ . Коэффициенты  $b_i$  будут выбираться посредством генетического алгоритма. В качестве подмоделей будем использовать сеть ANFIS с тремя входными переменными.

## 4.2 Результаты моделирования подмодели сети ANFIS

Рассмотрим вывод первой подмодели на обучающей выборке: синий цвет - реальные данные, красный цвет - подмодель 1



Рассмотрим вывод второй подмодели на обучающей выборке

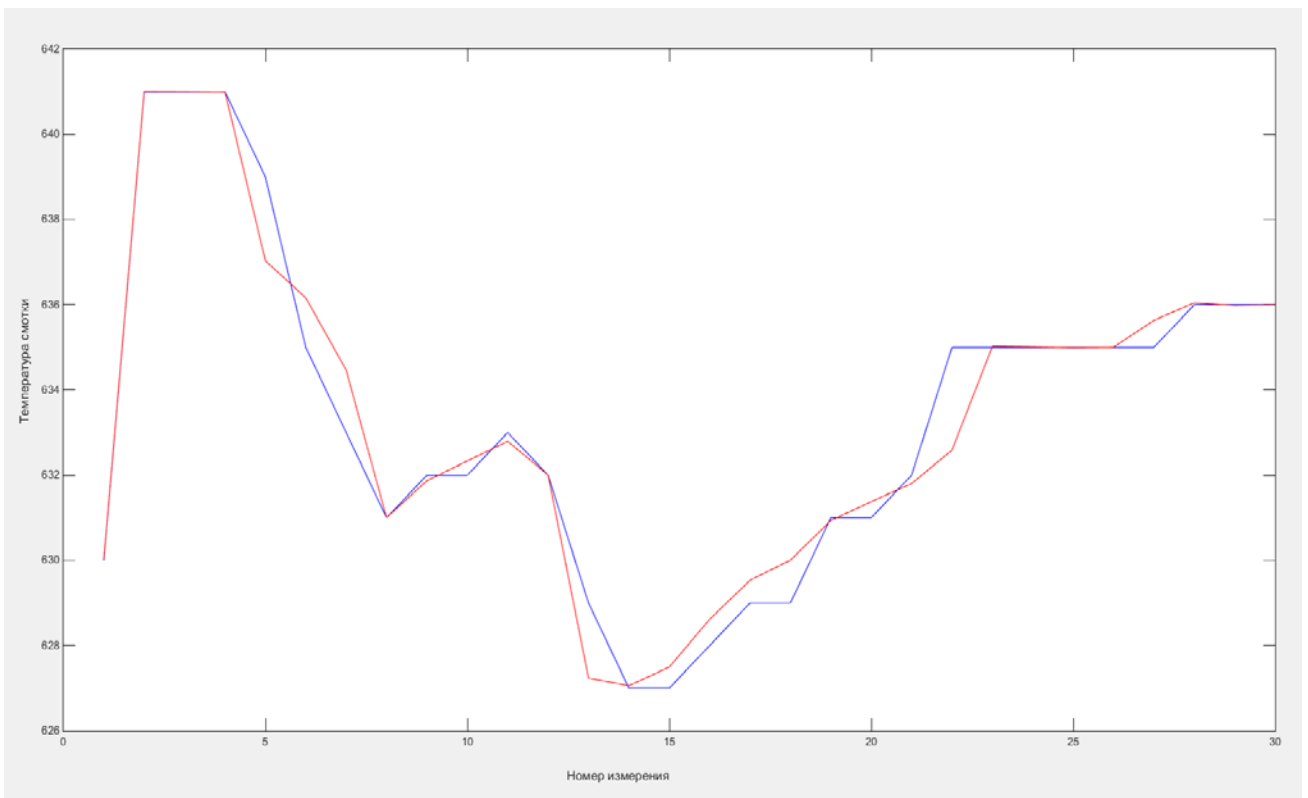
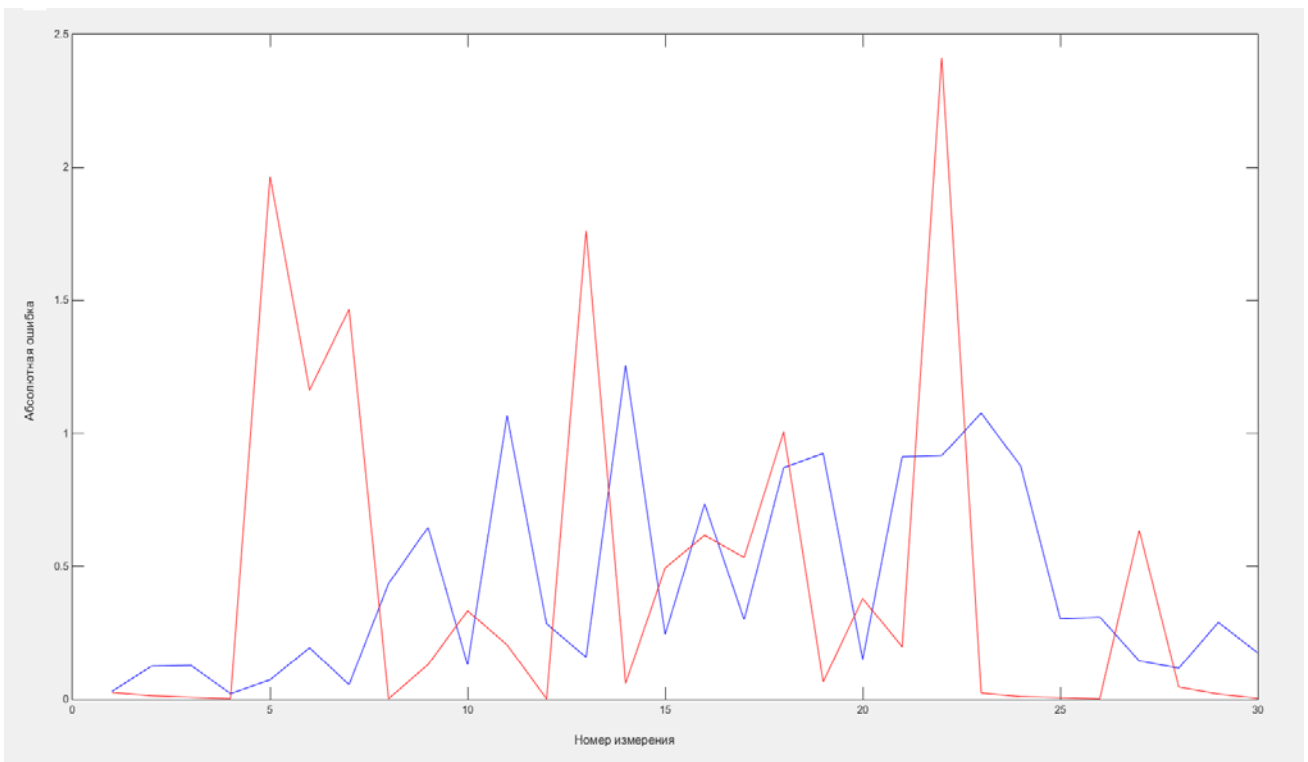


График абсолютных ошибок для обоих подмоделей: красный цвет - подмодель 1, синий цвет - подмодель 2



На тестовых данных абсолютная ошибка для обоих подмоделей не превышает значения в 2.5. Средняя абсолютная ошибка для первой

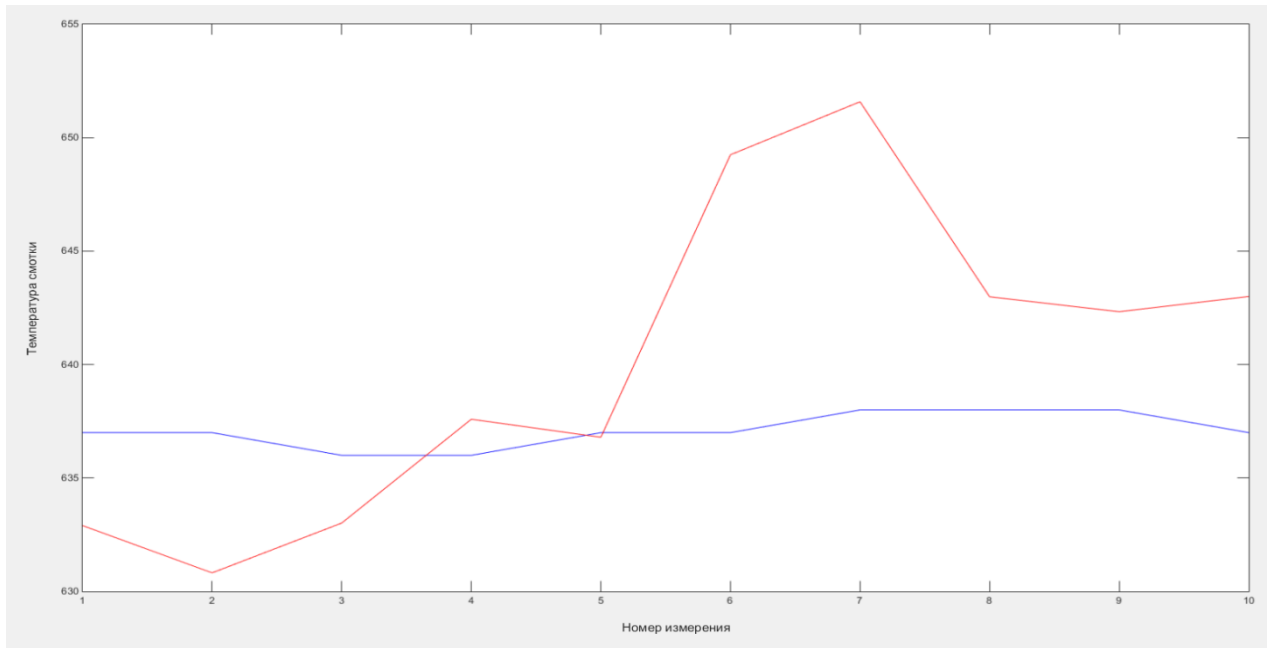


подмодели равна 0.431 для второй подмодели 0.453. Относительная ошибка

$\delta = \sum_{i=1}^n \frac{(x_i - \hat{x}_i)}{\hat{x}_i}$  для первой подмодели равна  $\delta = 0.0007048$  для второй

подмодели равна  $\delta = 0.0007156$ .

Рассмотрим вывод первой подмодели на тестовой выборке: красный цвет – подмодель 1, синий цвет – реальные данные



Рассмотрим вывод второй подмодели на тестовой выборке: красный цвет – подмодель 1, синий цвет – реальные данные

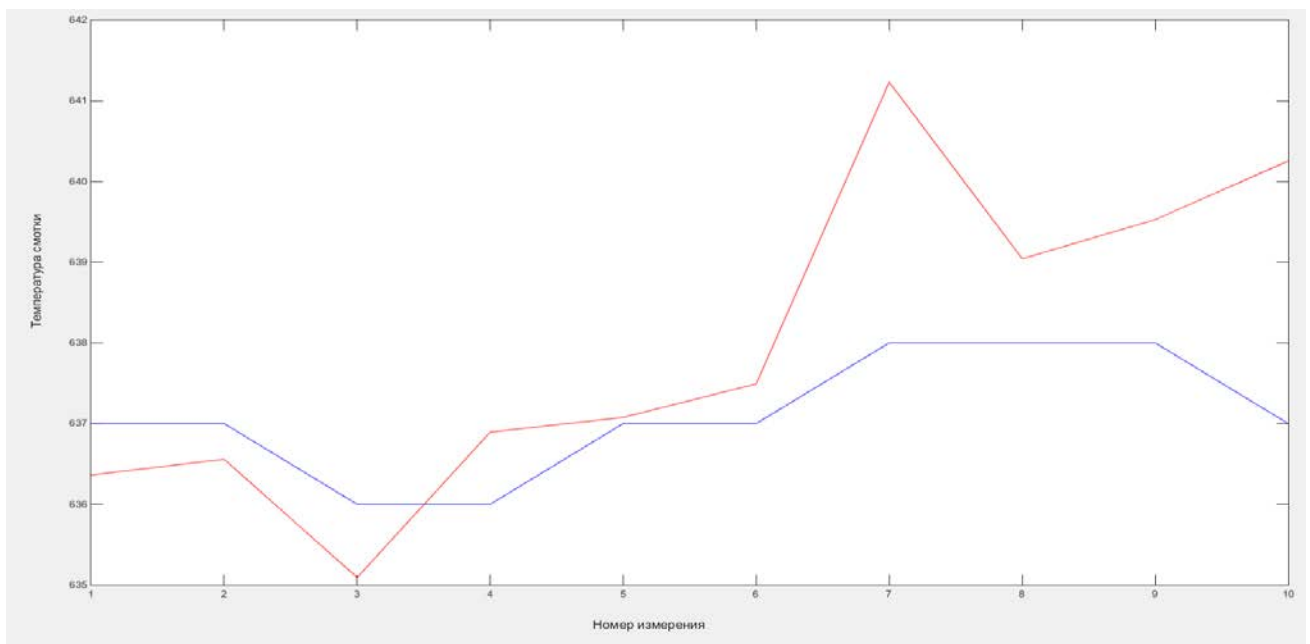
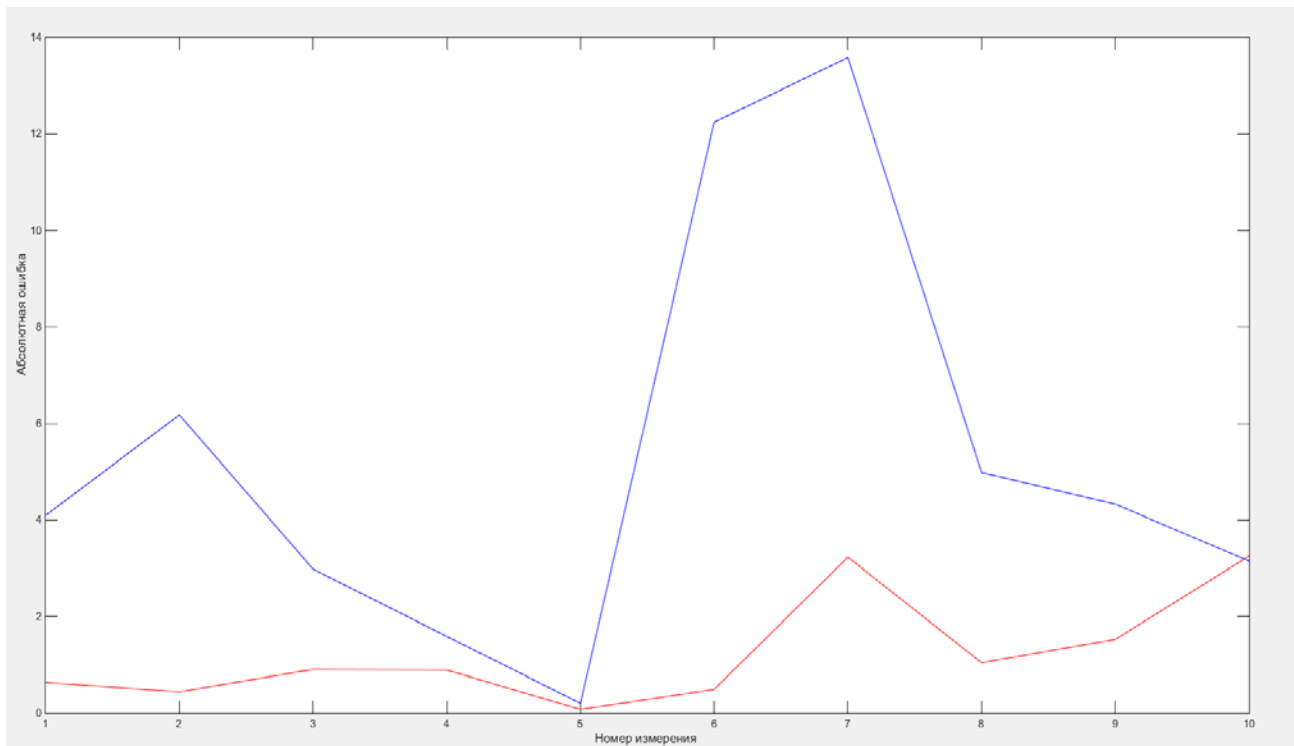


График абсолютных ошибок для обеих подмоделей: красный цвет - подмодель 2, синий цвет - подмодель 1



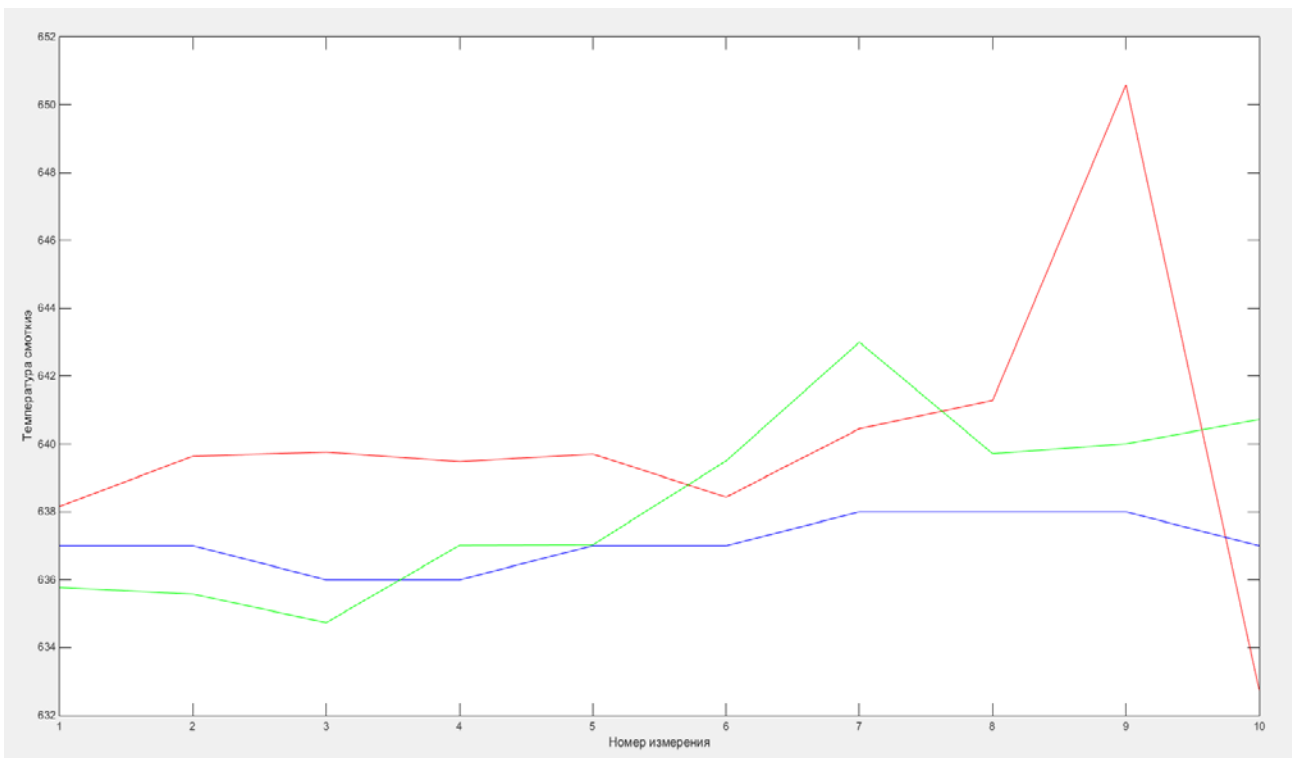
Максимальное значение абсолютной погрешности для первой подмодели 13.5 для второй подмодели 3.2. Среднее значение абсолютной погрешности для первой подмодели 5.3 для второй подмодели 1.25. Относительная ошибка  $\delta = \sum_{i=1}^n \frac{(x_i - \hat{x}_i)}{\hat{x}_i}$  для первой подмодели равна  $\delta = 0.008371262$  для второй подмодели равна  $\delta = 0.0019624432$ . Относительная ошибка для обеих подмоделей увеличилась на порядок, но осталась в пределах допустимого отклонения.

#### 4.3 Результаты моделирования душирующей установки

Объединим две подмодели методом двухступенчатой идентификации.

Результат объединения представлен на графике, для тестовой выборки:

синий – реальные данные, зеленый – объединенная модель, красный – полная модель



В результате для объединенной модели средняя абсолютная ошибка равна 1.91.

Относительная ошибка полной модели равна  $\delta = 0.094$ , для объединенной модели относительная ошибка равна  $\delta = 0.0696$

## 5 Построение модели котла

### 5.1 Описание модели котла

Рассматривался прямоточный паровой котел ТГМП-314 со следующими параметрами:

- производительность по свежему пару:  $D_{\text{пе}} = 1000$  т/ч;
- давление свежего пара за котлом:  $p_{\text{вых}} = 25$  МПа;
- давление вторично перегретого пара на входе:  $p'_{\text{вт.п}} = 4,03$  МПа;
- давление вторично перегретого пара на выходе:  $p_{\text{вт.п}} = 3,78$  МПа;
- температура питательной воды:  $t_{\text{вх}} = 270^\circ \text{C}$ ;
- температура вторично перегретого пара на выходе из котла:  $t_{\text{вых}} = 545^\circ \text{C}$ ;
- площадь станции:  $F_{\text{ст}} = 2167$  м<sup>2</sup>;

- площадь шахты:  $F_{ш} = 905/168 \text{ м}^2$ ;

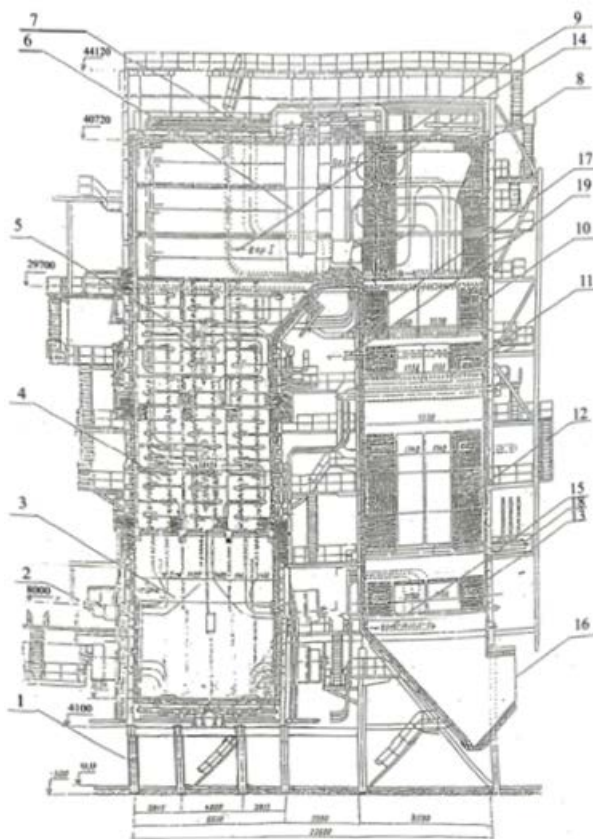
- площадь регенеративного воздухоподогревателя:  $F_{рВп} = 142200 \text{ м}^2$ .

Прямоточный котел — паровой котел, в котором полное превращение воды в пар происходит за время однократного прохождения воды через поверхность нагрева. Отличительной особенностью прямоточных котлов является отсутствие четкой фиксации экономайзерной и пароперегревательной зон. Прямоточный котел работают на докритическом и сверхкритическом давлении. Котел имеет П-образный профиль. П-образная компоновка — наиболее распространенная. В подъемной шахте располагается призматическая топочная камера, в опускной — конвективные поверхности нагрева. Преимуществом такой компоновки заключается в том, что тягодутьевые машины устанавливаются на нулевой отметке, что исключает вибрационные нагрузки на каркас котла. Основные недостатки возникают в связи с разворотом на 180 градусов возникают неравномерности омывания поверхности нагрева продуктами сгорания и концентрации золы по сечению конвективной шахты.

Верх топки экранирован фронтным топочным экраном и панелями экранов боковых стенок. Горелки расположены встречно в два яруса. Движение среды в экранах одноходовое. В горизонтальном газоходе расположен перегреватель сверхкритического давления. Он состоит из последовательно расположенных в газовом тракте ширм и пакета конвективного пароперегревателя. Тракт низкого давления пара состоит из двух пакетов промпароперегревателей. В опускном газоходе находится экономайзер. С котлом работают два регенеративных воздухоподогревателя.

В ПК вода с помощью питательного насоса подается в экономайзер, откуда поступает в змеевики, расположенные в топке. В выходной части змеевиков вода превращается в пар и начинается перегрев пара. Существенный недостаток ПК заключается в том, что, попадающие в котел с питательной водой соли, либо отлагаются на стенках змеевиков, либо вместе с паром

поступают в паровые турбины, где оседают на лопатках рабочего колеса, что снижает КПД турбины. Поэтому к качеству питательной воды для ПК предъявляются повышенные требования. Другой недостаток — увеличенный расход энергии на привод питательного насоса. Продольный разрез котла представлен на рис.26.



Продольный разрез котла Пп-1000-25-545/542-ГМ (ТГМП-314): 1 – каркас котла; 2 – газомазутные горелки (16 шт.); 3 – нижняя радиационная часть; 4 – СРЧ; 5 – ВРЧ; 6 – ширмовый пароперегреватель; 7 – потолочный экран топки и горизонтального газохода; 8 – экраны поворотной камеры; 9 – впрыскивающий пароохладитель первой ступени; 10 – конвективный пароперегреватель СКД; 11, 12 – конвективный пароперегреватель НД; 13 – экономайзер; 14 – пароохладитель второй ступени; 15 – вход питательной воды; 16 – выход дымовых газов; 17 – выход пара СКД; 18 – вход пара НД; 19 – выход пара НД

## 5.2 Результаты моделирования подмоделей сети ANFIS

Средняя абсолютная ошибка на тренировочных данных для первой подмодели равна 0.000726301, для второй подмодели 0.014967286.

Вывод модели для тестовых данных: синий цвет - реальные данные, красный цвет - первая подмодель, зеленый цвет - вторая подмодель

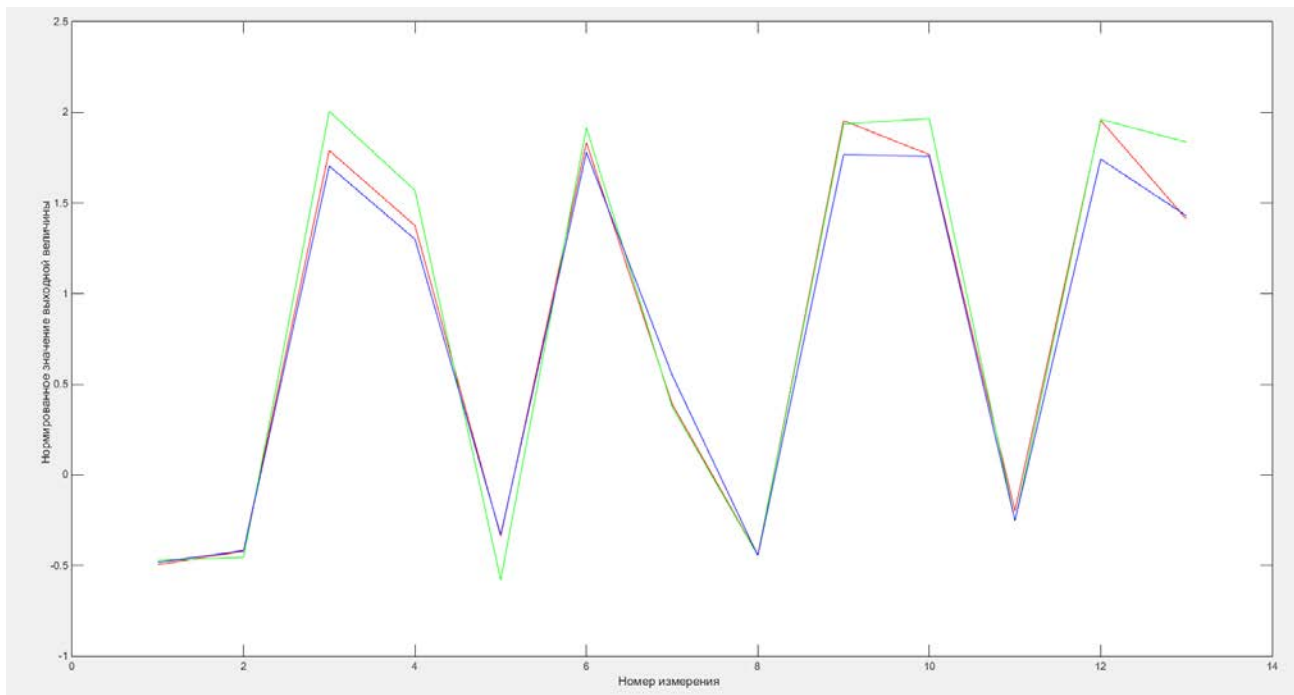
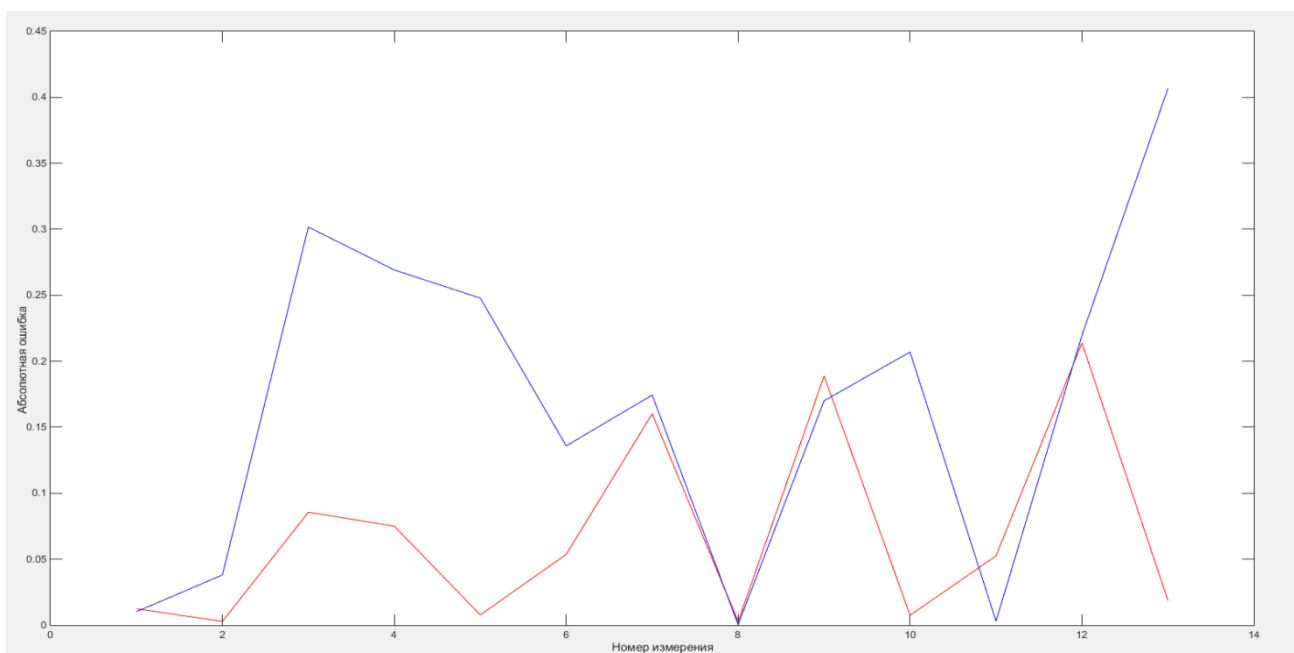


График абсолютной ошибки



Среднее абсолютной ошибки для первой подмодели 0.067863369

Среднее значение ошибки для второй подмодели 0.168050818

Относительная ошибка  $\delta = \sum_{i=1}^n \frac{(x_i - \hat{x}_i)}{\hat{x}_i}$  для первой подмодели равна

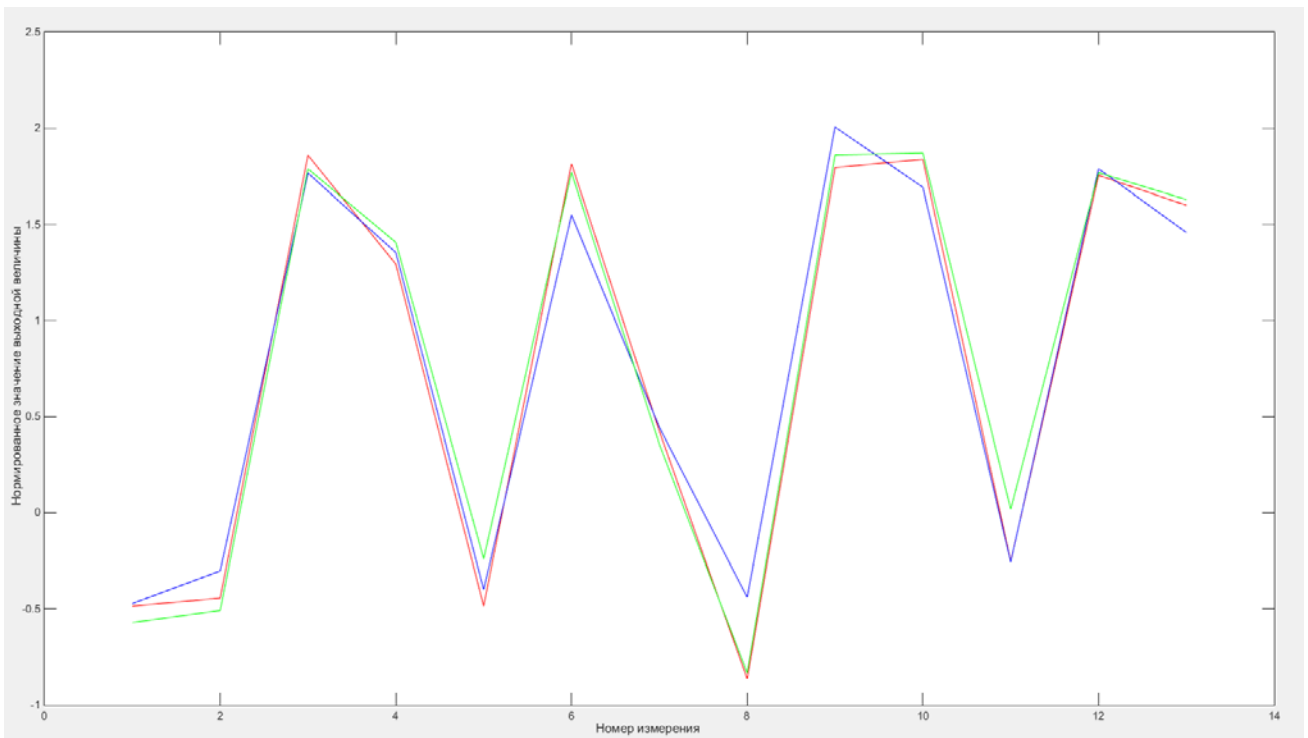
$$\delta = 0.0729842$$

Относительная ошибка  $\delta = \sum_{i=1}^n \frac{(x_i - \hat{x}_i)}{\hat{x}_i}$  для второй подмодели равна

$$\delta = 0.175227092$$

### 5.3 Результаты моделирования котла

Синий – реальные, красный – объединение двух моделей по 5 входов в каждой, зеленый – модель 10 входов



В итоге для объединенной модели относительная ошибка  $\delta = \sum_{i=1}^n \frac{(x_i - \hat{x}_i)}{\hat{x}_i}$

равна  $\delta = 0.05371$ . Для полной модели с 10 входами относительная ошибка

равна  $\delta = 0.1394$

## **6 Выводы**

В работе описано современное состояние интеллектуальных методов моделирования систем, сделан обзор алгоритмов моделирования систем на основе нечеткой логики и нейронных сетей. Предложен генетический алгоритм моделирования систем, основанный на методе многоступенчатой идентификации. Рассмотрены вопросы оценки адекватности математических моделей.

Полученные нейро-нечеткой модели и алгоритмы были использованы для моделирования душирующей установки и парового котла ТГМП-314 теплоэлектростанции. В процессе моделирования использован метод многоступенчатой идентификации, включающий в себя полную модель и подмодели процесса. В работе приводится сравнение полной модели и модели, построенной с использованием генетического алгоритма, основанного на методе двухступенчатой идентификации. Обе модели показали хорошую точность, удовлетворяющую допустимым ограничениям на ошибку моделирования на тестовых данных. На примерах моделирования душирующей установки и котла по экспериментальным данным показана эффективность использованных моделей для систем с запаздыванием. Относительная ошибка для модели, построенной с использованием генетического алгоритма, равна  $\delta = 0.0537$  для котла,  $\delta = 0.0696$  для душирующей установки. Для полной модели, основанной на сети ANFIS, использующей метод наименьших квадратов и метод обратного распространения ошибки, относительная ошибка для котла равна  $\delta = 0.1394$ , для душирующей установки  $\delta = 0.0945$ .



## **7. Список литературы**

1. Борисов В.В. Круглов В.В. Федулов А.С. Нечеткие модели и сети
2. Кофман А. Введение в теорию нечетких множеств. – М.: Мир, 1982. 432с.
3. Kudinov Y.I., Kudinov I.Y., Pashchenko F.F. Identification of Multivariable Fuzzy Systems. Proceedings of the 2011 Chinese Control and Decision Conference (CCDC) 23-25 May 2011 Mianzhou Hotel, Mianyang, China, pp. 1572-1575
4. Рутковская Д. Пилиньский М. Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы.
5. Пащенко Ф.Ф. Введение в состоятельные методы моделирования систем. Ч.1 Математические основы моделирования систем. 2006, 328с.
6. Пащенко Ф.Ф. Введение в состоятельные методы моделирования систем. Ч.2 Идентификация нелинейных систем. 2007, 288с
7. Ajith Abraham, Neuro Fuzzy Systems: state of Art Modelling Techniques, In proceedings of the sixth international work conference on Artificial and

- Natural Neural Networks, IWANN 2001, Granada, Springer Verlag Germany, pp. 269–276, June 2001.
8. Azadeh, A. Nov. 2006 Integration of Artificial Neural Networks and Genetic Algorithm to Predict Electrical Energy consumption
  9. Berenji H. R., Khedkar P., Learning and tuning fuzzy logic controllers through Reinforcements, IEEE Transactions on Neural Networks, 1992, vol.3,nr.5,724-740
  10. Hayashi I., Nomura H., Yamasaki N., On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm, IEEE Transactions on Neural Networks, 1992, vol.6,241-246
  11. Juang Chia Feng, Lin Chin Teng, An Online Self Constructing Neural Fuzzy Inference Network and its Applications, IEEE Transactions on Fuzzy Systems, Vol 6, № 1, pp. 12–32, 1998
  12. Kazakidis, S.A. Oct. 2012 Electricity consumption prognosis with the combination of smart metering and artificial neural networks
  13. Langton, C., Genetic Algorithms and Artificial Life, MIT Press
  14. Latif, M.S., Hong Zhou, Amir, M., A hybrid Quantum Estimation of Distribution Algorithm (Q-EDA) for Flow-Shop Scheduling, Natural Computation (ICNC), 2013 Ninth International Conference, 23-25 July 2013, 654 – 658
  15. Liu, Jiabin Aug. 2012 An improved genetic algorithm to minimal test cost reduction
  16. Lin C-T, Lee G Neural-network-based fuzzy logic control and decision system. IEEE Transactions on Computers, December 1991, p 1320-1336
  17. Mohammad Zadeh, S. July 2010 Modeling residential electricity demand using neural network and econometrics approaches
  18. Nagothu, K.S. June 2013 Improved genetic and Harmony Search Algorithms for single and multi stage expansion planning of power system
  19. Nasiraghdam, M. March 2010 Query optimization in distributed database using hybrid evolutionary algorithm

20. Revello, T.E., McCartney, R., Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress, 1086 – 1091
21. Ono, K. June 2013 Island model genetic programming based on frequent trees
22. Saravanan, S. March 2014 India's electricity demand estimation using Genetic Algorithm
23. Tadeusiewicz R., Sieci neuronowe, Akademia Oficyna Wydawnicza RM, Warszawa 1993
24. Trivedi, N., Wei Lai, Zhongwei Zhang, Optimizing windows layout by applying a genetic algorithm, Evolutionary Computation, 2001. Proceedings of the 2001, 431 - 435 vol. 1
25. Zne-Jung Lee, A hybrid algorithm applied to travelling salesman problem, Networking, Sensing and Control, 2004 IEEE International Conference on Volume:1 , 21-23 March 2004
26. Zhang Xing-ping Nov. 2008 Electricity Consumption Forecasting Based on Improved BP Neural Network

### 8 Приложение 1

N(i)	N(i-1)	V(i)	V(i-1)	H(i)	T конца прокатки	T смотки
5	5	790	790	1,81	777	630
8	5	792	790	1,83	792	641
9	8	794	792	1,85	795	641
10	9	796	794	1,82	805	641
10	10	798	796	1,82	802	639
10	10	799	798	1,84	799	635
10	10	801	799	1,78	797	633
9	10	806	801	1,78	796	631
9	9	807	806	1,79	795	632
9	9	809	807	1,8	794	632
9	9	810	809	1,81	795	633
10	9	812	810	1,82	794	632
10	10	813	812	1,81	795	629
10	10	817	813	1,81	795	627

10	10	819	817	1,81	797	627
10	10	822	819	1,8	798	628
10	10	824	822	1,78	799	629
10	10	825	824	1,8	799	629
10	10	827	825	1,8	799	631
10	10	828	827	1,79	798	631
10	10	829	828	1,84	796	632
10	10	831	829	1,84	797	635
11	10	834	831	1,82	795	635
11	11	836	834	1,82	796	635
11	11	838	836	1,8	796	635
10	11	841	838	1,78	798	635
10	10	842	841	1,75	798	635
10	10	844	842	1,77	803	636
11	10	845	844	1,77	803	636
11	11	847	845	1,78	800	636
11	11	849	847	1,8	801	637
11	11	850	849	1,81	801	637
12	11	851	850	1,82	800	636
12	12	854	851	1,82	799	636
12	12	855	854	1,81	799	637
12	12	857	855	1,81	797	637
11	12	859	857	1,8	796	638
11	11	861	859	1,79	797	638
11	11	863	861	1,77	795	638
11	11	866	863	1,82	794	637

## 9 Приложение 2

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Y
-1,139	0,315	1,286	1,288	0,151	0,151	-0,859	-0,989	0,916	0,960	-0,375
-1,139	0,315	1,286	1,288	0,151	0,151	-0,859	-0,989	0,916	0,960	-0,375
0,661	-1,389	0,431	-0,718	0,151	0,151	1,448	1,531	-0,336	-1,086	1,573
-0,059	0,376	0,859	0,519	0,151	0,151	-0,282	0,868	0,535	0,358	0,059
-0,779	1,210	1,445	1,458	0,151	0,151	-1,205	-1,055	1,161	1,080	-0,503
-0,779	0,654	1,185	1,114	0,151	0,151	-1,205	-0,790	1,039	0,960	-0,341
-0,779	0,412	1,123	0,944	0,151	0,151	-0,974	-0,723	0,957	0,719	-0,260
-0,779	0,739	1,342	1,232	0,151	0,151	-0,628	-0,922	0,780	0,840	-0,367
-0,779	1,379	1,602	1,520	0,151	0,151	-1,089	-1,188	0,971	1,080	-0,548
-0,959	0,062	0,679	0,472	0,151	0,151	-0,628	-0,723	0,712	0,599	0,037
-0,959	1,258	1,470	1,232	0,151	0,151	-0,628	-0,922	0,780	0,840	0,068
-0,059	-1,027	0,693	-0,420	0,151	0,151	0,295	0,669	0,304	-0,965	0,832
-0,959	-1,292	0,571	-0,307	0,151	0,151	0,525	0,536	0,304	-0,845	0,769
-0,779	1,295	1,340	1,473	0,151	0,151	-0,974	-1,121	1,012	1,080	-0,527
1,382	-1,147	0,344	-0,760	0,151	0,151	0,756	1,266	-0,485	-1,326	1,763
1,022	-1,365	0,330	-0,770	0,151	0,151	0,756	1,266	-0,485	-1,326	1,763
1,022	2,637	1,152	1,109	0,151	0,151	-0,513	-0,856	0,345	0,719	-0,276
1,022	-1,075	0,330	-0,619	0,151	0,151	0,987	1,067	-0,472	-1,086	1,486
0,301	-0,253	0,543	-0,010	0,151	0,151	-0,051	0,139	0,399	0,118	0,524
-0,779	0,533	1,237	1,180	0,151	0,151	-0,974	-0,856	0,318	0,599	-0,300
1,382	-1,087	0,520	-0,704	0,151	0,151	1,217	1,266	-0,703	-1,326	1,701
-0,419	-0,156	0,867	0,453	0,151	0,151	-0,282	-0,326	-0,172	0,238	0,131
-1,139	0,726	1,476	1,468	0,151	0,151	-0,859	-0,989	0,780	0,960	-0,576
1,022	-1,329	0,584	-0,860	0,151	0,151	1,448	1,266	-0,771	-1,086	1,695
-0,419	-0,277	0,811	0,354	0,151	0,151	-0,167	-0,127	0,005	0,238	0,193
-1,499	0,231	1,234	1,496	0,151	0,151	-0,743	-1,121	0,862	1,201	-0,506
-0,059	-0,724	0,667	-0,203	0,151	0,151	0,179	0,271	0,318	-0,243	0,626
-0,779	0,751	1,263	1,237	0,151	0,151	-0,743	-0,856	1,012	0,840	-0,371
-0,779	0,521	1,158	1,071	0,151	0,151	-0,743	-0,790	0,835	0,840	-0,297
-1,139	0,630	1,210	1,440	0,151	0,151	-1,089	-1,121	0,943	0,960	-0,476
-0,779	1,137	1,208	1,425	0,151	0,151	-1,089	-1,121	0,889	0,960	-0,482
-0,779	0,908	1,406	1,317	0,151	0,151	-0,974	-0,989	0,862	0,960	-0,418
1,382	-1,087	0,596	-0,864	0,151	0,151	1,679	1,332	-1,057	-1,567	1,705
1,022	-0,857	0,557	-0,614	0,151	0,151	1,333	1,067	-0,744	-1,086	1,301
-1,139	0,182	1,117	1,180	0,151	0,151	-0,859	-0,790	0,263	0,840	-0,330
1,742	-0,954	0,489	-0,888	0,151	0,151	1,679	1,332	-1,057	-1,326	1,781
-0,059	-0,615	0,530	-0,057	0,151	0,151	0,410	0,271	-0,091	0,238	0,551
-0,239	1,476	1,487	1,407	0,151	0,151	-1,089	-1,055	0,889	0,960	-0,444
1,742	-0,942	0,615	-0,869	0,151	0,151	1,448	1,332	-1,125	-1,567	1,766
1,382	-1,147	0,524	-0,793	0,151	0,151	1,333	1,465	-1,261	-1,567	1,760
-0,419	0,823	1,197	1,048	0,151	0,151	-0,859	-0,723	0,127	0,719	-0,250
1,742	-0,918	0,635	-0,845	0,151	0,151	1,794	1,266	-1,424	-1,326	1,743
0,661	-1,244	0,508	-0,652	0,151	0,151	1,333	1,266	-1,247	-1,447	1,430

