

УДК 004.23, 004.72

## ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ FUZZY SMART SENSORS WSN ДЛЯ ИНТЕЛЛЕКТУАЛИЗАЦИИ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЯХ

Аверкин Алексей Николаевич<sup>1</sup>, Лавров Георгий Константинович<sup>2</sup>

<sup>1</sup>Кандидат физико-математических наук, доцент;  
ГБОУ ВО МО «Университет «Дубна»,  
Институт системного анализа и управления;  
141980, Московская обл., г. Дубна, ул. Университетская, 19;  
e-mail: averkin2003@inbox.ru.

<sup>2</sup>Аспирант;  
ГБОУ ВО МО «Университет «Дубна»,  
Институт системного анализа и управления;  
141980, Московская обл., г. Дубна, ул. Университетская, 19;  
e-mail: glavrov@mail.ru.

*В статье предлагается к рассмотрению технология построения беспроводных сенсорных сетей, основанных на нечетких сенсорных узлах, представлен ее инструментарий. Подробно разобраны компоненты технологии, приведены примеры возможного использования. В заключении даны выводы о перспективах технологии.*

**Ключевые слова:** беспроводная сенсорная сеть, умный сенсорный узел, нечеткий контроллер, система поддержки принятия решений.

## USING THE FUZZY SMART SENSORS WSN TECHNOLOGY FOR INTELLECTUALIZATION OF DECISION SUPPORT SYSTEMS IN WIRELESS SENSOR NETWORKS

Averkin Alexey<sup>1</sup>, Lavrov Georgy<sup>2</sup>

<sup>1</sup>Candidate of Science in Physics and Mathematics, assistant professor;  
Dubna State University,  
Institute of system analysis and management;  
141980, Dubna, Moscow reg., Universitetskaya str., 19;  
e-mail: averkin2003@inbox.ru.

<sup>2</sup>PhD student;  
Dubna International University of Nature, Society and Man,  
Dubna State University,  
141980, Dubna, Moscow reg., Universitetskaya str., 19;  
e-mail: glavrov@mail.ru.

*The article proposes to consider the technology of construction of wireless sensor networks based on smart sensor nodes and represents its tools. The technology components are discussed in detail, examples of possible use are described. In conclusion, the findings on technology prospects are given.*

**Keywords:** wireless sensor network, smart sensor node, fuzzy controller, decision support system.

### Введение

Предлагаемая технология *Fuzzy Smart Sensors WSN* (далее – *Smart Sensors*) основана на использовании методов нечеткой логики для интеллектуализации беспроводных сенсорных сетей (БСС). БСС в настоящее время является одной из наиболее перспективных информационных технологий 21-го века. Впервые «умные» сенсоры, или моты, были реализованы совместно Калифорнийским университетом в Беркли и корпорацией Интел. Прототип узла БСС (мота) представляют собой программно-аппаратную платформу для установки набора специализированных сенсоров на основе автономного

беспроводного контроллера. Для обеспечения возможности самоорганизации мотов в сенсорную сеть используется специализированная операционная система *TinyOS*, а также программная среда разработки приложений. БСС представляет собой большое количество миниатюрных устройств, помещенных в физическую среду и функционирующих, как единая сеть. *TinyOS* обеспечивает необходимые для сенсорной среды возможности коммуникации, маршрутизации и поддержки приложений.

Одной из главных задач является интеллектуализация БСС за счет предварительной обработки информации для того, чтобы не посылать пользователю сырую или ошибочную информацию, осуществлять кластеризацию, агрегацию и слияние разнородных данных, осуществлять распределенную обработку сложных запросов внутри сети и осуществлять маршрутизацию и оптимизацию расхода энергии. Методы интеллектуализации принятия решений, основанные на применении моделей классического символического искусственного интеллекта, являются малоприменимыми из-за ограниченности вычислительных и энергетических ресурсов БСС. В настоящее время для решения этих задач используется подход нечеткой логики, представляющий собой набор методологий обработки информации, позволяющих работать с неточными, неопределенными и нечеткими данными и обеспечивающих построение гибких и интерпретируемых моделей, использующих субъективную экспертную информацию. Подобный подход является обобщением на случай вложенных, распределенных, адаптивных систем концепции мягких измерений. Мягкие измерения возникли на основе гибридизации методов мягких вычислений и математической статистики для обработки результатов разнородных измерений [5]. Необходимость использования таких методов для БСС связана еще и с тем, что с настоящее время практически отсутствуют точные аналитические модели обработки информации в БСС.

Наиболее удобным для построения интеллектуальных систем распознавания и поддержки принятия решений на основе сенсорных сетей является модель нечеткого (мягкого, виртуального) сенсора [2], ядром которого является машина вывода, представляющая собой набор нечетких продукционных правил с блоками фаззификации и дефаззификации, осуществляющих перевод численных измерений в лингвистическую форму и наоборот. Выходом такого сенсора может быть любая функция входных параметров, например, параметры, которые трудно или невозможно измерить, результаты агрегации или слияния входных параметров, управляющие воздействия на сеть для маршрутизации или оптимизации работы сети. Построение базы знаний нечеткого сенсора может быть осуществлено как экспертом, так и в результате самообучения.

В настоящее время эти методы реализуются на базе разрабатываемого семейства *Smart Sensors*, представляющего собой инструментарий на базе программных компонент, алгоритмов и аппаратных решений, позволяющих осуществлять быструю разработку прикладных беспроводных сенсорных сетей. Семейство *Smart Sensors* предназначено для автоматизации зданий и производственных процессов, систем безопасности, поддержания жизнеобеспечения, экологического мониторинга, а также систем диагностики и сигнализации.

Таким, образом настоящее время разрабатываются следующие компоненты семейства *Smart Sensors*:

- *Smart Sensors Platform* – программно-аппаратная платформа, содержащая программные компоненты, алгоритмы, аппаратные реализации для следующего поколения *M2M*;
- *Smart Sensors Cube* – узел БСС, состоящий из платы микроконтроллера, платы коммуникации данных, интерфейсной платы, платы беспроводной связи, сенсоров и актуаторов, а также системного программного обеспечения, основанное на *TinyOS*;
- *Smart Sensors expert* – вложенная в БСС программная среда, позволяющая осуществлять удаленный мониторинг и управление процессами и объектами;
- *Fuzzy Smart Sensors expert* – реализация *Smart Sensors expert* на основе вложенных в узлы БСС *Fuzzy Sensors*, позволяющая поддерживать функции распределенной нечеткой экспертной системы на узлах БСС;
- *Fuzzy Sensor* – реализация нечеткого сенсора на одном *Smart Sensors Cube*.

В рамках технологии *Smart Sensors* на основе аппарата нечетких вложенных систем реализована универсальная оболочка нечеткого сенсора *Fuzzy Sensor*, вложенная во все узлы БСС, которая, может аппроксимировать широкий класс из существующих алгоритмов слияния и агрегации данных. Каждый узел сети, помимо функций сбора и передачи информации, может быть запрограммирован на реализацию любой необходимой зависимости между входными и выходными переменными. В эти

вершины по сети могут пересылаться параметры нечетких сенсоров – шаблоны правил (номера переменных и термов, входящие в левую и правую часть или параметры новых термов, параметры триангулярных норм). В ряде задач, например, слежение за целью, такие пересылки могут осуществляться вершинами самостоятельно, на основании имеющихся правил, в зависимости от положения цели и содержания запроса. Однако в более сложных случаях используется еще один уровень вершинкоординаторов, которые управляют пересылкой знаний по сети или самообучением вершин и сети и, в общем случае, обладают данными о распределении знаний по сети, у них имеются внутренние модели среды и наблюдаемых объектов.

Разработанный универсальный инструментарий для систем интеллектуальной поддержки принятия решений в БСС на базе *Smart Sensors* может быть использован для следующих уровней обработки данных:

- сетевой уровень (маршрутизация, энергосбережение, активные сети);
- уровень баз данных (нечеткий язык запросов);
- уровень распределенной поддержки принятия решений (слияние и агрегация данных);
- уровень распределенной базы знаний (распределенной ЭС);
- уровень многоагентной системы (реактивные агенты).

## 1. Проблемы обработки данных в БСС, требующие интеллектуализации сенсоров

Основной целью функционирования БСС является прямой сбор громадного количества данных и передача их пользователю. Собранные данные могут интерпретироваться как распределенная база данных. При таком подходе, как у пользователя, так и у разработчика возникают проблемы, связанные с организацией сильно распределенных процессов обработки неточных, неполных, недоопределенных или, наоборот, избыточных сенсорных данных.

Основными факторами, сильно влияющими на функционирование БСС и усложняющими их проектирование, являются следующие:

- узлы БСС обладают крайне ограниченными вычислительными возможностями;
- радиус коммуникации между узлами в большинстве приложений крайне ограничен и в ряде случаев ограничен ближайшими соседями;
- БСС содержит большое количество ненадежных узлов, допускающих ошибки при измерении и при передаче информации;
- БСС должна надежно функционировать при разрушении большей части ее узлов;
- вершины или их отдельные каналы связи могут временно прекращать функционировать из-за изменений внешней среды и внутреннего состояния;
- в силу вышеупомянутых причин топология БСС может постоянно меняться и включать множество дублирующих маршрутов;
- расход энергии в узлах крайне неравномерен;
- энергетические и временные затраты на передачу данных увеличиваются с увеличением размеров сети.

Две последних проблемы представляют для пользователя особую важность. Пропускная способность сети лишь незначительно увеличивается в зависимости от числа вершин (увеличивается, как  $\log N$ , где  $N$  – число вершин). Наиболее целесообразным выходом из этой ситуации является уменьшение потока данных через сеть за счет добавления распределенной иерархической обработки данных внутри сети и передачи пользователю лишь релевантных ответов на запрос. Существует целый спектр разнообразных алгоритмов для реализации этой обработки, зависящих от типа данных и уровня их обобщения. Но в традиционных моделях распределенной иерархической обработки каждый алгоритм, как правило, жестко связывается с соответствующей вершиной в рамках заданной топологии сети. При изменении топологии сети необходимо перепрограммирование вершин, что связано с пересылкой по сети больших объемов программного кода. Для решения данной проблемы в *Smart Sensors* предлагается заменить пересылку данных или программного кода пересылкой знаний, нужных для необходимой настройки локальных узлов.

Предпоследняя проблема (неравномерность расхода энергии) нами решается введением соответствующих правил в заданный сенсорный узел для возможности получения им автономного решения о целесообразности его участия в сборе и передаче данных при данном состоянии среды, соседних узлов и узла, принимающего решение.

В качестве инструментария для решения этих проблем нами реализована универсальная оболочка нечеткого сенсора с продукционной моделью представления знаний, вложенная во все узлы БСС - *Fuzzy Sensor*. Эта оболочка может аппроксимировать широкий класс из существующих алгоритмов слияния и агрегации данных. При таком подходе каждый сенсор становится интеллектуальным агентом, обладает знаниями о себе и о среде и способен к самостоятельному принятию решений. У сенсора возникает возможность управления этими знаниями и их передачи другому сенсору. БСС в этом случае может быть интерпретирована как распределенная БД и БЗ с возможностями мобильности и адаптивности. Это позволяет использовать при создании БСС технологии многоагентных систем и распределенных интеллектуальных систем поддержки принятия решений.

## 2. Smart Sensors как технология вложенных нечетких систем

### 2.1. Нечеткая машина вывода

В качестве инструментария для создания универсальной вложенной оболочки для интеллектуальной поддержки принятия решений предлагается нечеткая машина вывода, представляющую собой нечеткий контроллер, и позволяющая, согласно результатам У.Ванга (1992) и К.Кастро (1995) [3], в случае треугольных или гауссовых функций принадлежности нечетких множеств, аппроксимировать любую зависимость «вход-выход» в виде набора нечетких правил типа  $R^{(k)}$ ,  $k = 1, \dots, N$ .

$R^{(k)}$ : ЕСЛИ ( $x_1$  это  $A^k_1$  И  $x_2$  это  $A^k_2$  ... И  $x_n$  то  $A^k_n$ ) ТО ( $y_1$  это  $B^k_1$  И  $y_2$  это  $B^k_2$  ... И  $y_n$  то  $B^k_n$ ),

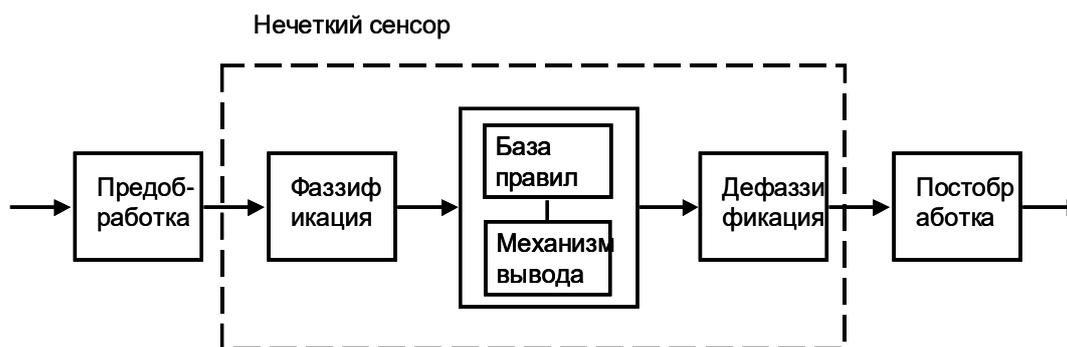
где  $N$  – количество нечетких правил,  $x_i$  и  $y_j$  – входные и выходные переменные лингвистической модели,  $A^k_i$ ,  $B^k_j$  – нечеткие множества,  $i = 1, \dots, n$ ,  $j = 1, \dots, m$ . Например, пусть  $x_1$  – температура,  $x_2$  – влажность,  $x_3$  – освещенность,  $y_1$  – комфортность,  $A^k_i$ ,  $B^k_j$  – нечеткие лингвистические значения температуры, влажности, освещенности и комфортности соответственно по шкале (ОЧЕНЬ НИЗКАЯ, НИЗКАЯ, СРЕДНЯЯ, ВЫСОКАЯ, ОЧЕНЬ ВЫСОКАЯ), то можем записать следующие правила для вычисления комфортности.

$R^{(1)}$ : ЕСЛИ (температура – СРЕДНЯЯ И влажность – ВЫСОКАЯ И освещенность – ВЫСОКАЯ), ТО (комфортность ОЧЕНЬ ВЫСОКАЯ)

$R^{(2)}$ : ЕСЛИ (температура – НИЗКАЯ И влажность – ВЫСОКАЯ И освещенность – НИЗКАЯ), ТО (комфортность – НИЗКАЯ).

На вход машины вывода поступают численные значения температуры, влажности, освещенности, на выход – численное значение комфортности в некотором виртуальном интервале, например,  $[0,10]$ , где 0 соответствует минимальному возможному значению комфортности, а 10 – максимальному.

Машина вывода представляет собой совокупность базы нечетких продукционных правил и механизма вывода с блоками фаззификации и дефаззификации, осуществляющими перевод численных измерений в лингвистическую форму и наоборот (рис. 1). Выходом машины вывода может быть любая функция входных параметров, например, параметров, которые трудно или невозможно измерить, результаты агрегации или слияния входных параметров, управляющие воздействия на сеть для маршрутизации или оптимизации работы сети.



*Рис. 1. Нечеткая машина вывода*

Данная модель обладает следующими качествами, которые обусловили ее выбор для реализации на сенсорной сети:

- универсальность;
- адаптивность;
- возможность создания иерархических конфигураций;
- мобильность;
- возможность программирования на языке, близком к ЕЯ;
- небольшой объем кода;
- простота создания пользователем собственных приложений.

## 2.2. Фаззификация

Фаззификатор отображает вход на область распределений возможности и имеет следующие параметры:

- число функций принадлежности;
- форму функций принадлежности (например, треугольная, гауссова и т.д.).

Оценка центра функции (например, центр масс) и рассеивания (например, стандартное отклонение или диапазон изменения) функций принадлежности.

## 2.3. Блок вывода

Блок вывода – это блок принятия решений системы, который определяет, как система интерпретирует нечеткие лингвистические переменные. Его параметрами являются операторы агрегации, которые являются обобщением классической интерпретации связей «ИЛИ» и «И» - MAX и MIN.

## 2.4. Дефаззификация

На этапе дефаззификации нечеткие консеквенты отображаются в четкие выходные величины. На его результат влияют:

- число функций принадлежности;
- форма функций принадлежности.

Форма нечеткой импликации, так как величина консеквента из машины вывода формирует выходную функцию принадлежности перед дефаззификацией. Мера оценки центра консеквента изменяет выходные функции принадлежности. Обычно используется центр масс, хотя медианы и моды также используются для получения четкого значения.

Фаззификация и дефаззификация требуют выбора размерности, положения и формы функций принадлежности. Операция дефаззификации сама может быть параметризована, как и блок вывода, зависящий от выбора операторов агрегации, которые также могут быть параметризованы. Все эти параметры могут быть адаптивно настроены с помощью меры достижения цели в некоторой среде обучения с учителем. Для оптимизации функционирования БСС предлагается следующие способы

адаптации нечетких функций принадлежности, баз правил и операторов агрегации: использование самоорганизующегося нечеткого сенсора, который предполагает возможность генерации и адаптации правил как с учителем, так и без учителя; численные методы, включающие метод наискорейшего спуска на параметрах центра тяжести и ширины носителей.

## 2.5. Настройка операторов агрегации

Необходимость в такой настройке возникает, когда эксперт, заполняющий базу правил, удовлетворен как числом, так и формой функций принадлежности, так же, как и самими правилами. Используются обобщенные операторы для реализации нечетких операций пересечения и объединения –  $T$ -нормы (для пересечения) и  $T$ -конормы (для объединения). Эффективность машины вывода сильно зависит от выбора  $T$ -операторов для данной проблемы.

Важным свойством предлагаемой архитектуры и ее программно-аппаратной реализации в БСС является поддержка набора различных правил в рамках одной и той же программы, позволяющей осуществлять такие независимые нечеткие алгоритмы, как управление (узлом, сетью), мониторинг (среды, объекта), и непосредственно измерение (реальных и виртуальных атрибутов), реализованные на одном и том же устройстве. Это особенно полезно в приложениях, когда используются сравнительно дешевые сенсоры с низкой разрешающей способностью. Тогда нечеткие сенсоры значительно удешевляют стоимость измерений, выступают в качестве «виртуальных» дорогих или отсутствующих в сети сенсоров. Таким образом, нечеткие сенсоры выступают в качестве отображений, которые связывают атрибуты, для которых в обычной ситуации требуется специальный сенсор, с переменными, которые могут быть измерены существующими дешевыми дополнительными сенсорами. При мониторинге холодильной установки, например, могут быть запрошены данные о скорости воздушного потока и скорости вращения вентилятора. Добавление специальных сенсоров потребует времени и увеличит стоимость сети. Однако эти значения могут быть с достаточной точностью получены с помощью имеющихся сенсоров температуры воздуха и напряжения в сети электромотора в виде набора продукционных правил. Для получения соответствующих нечетких отображений нет необходимости в наличии формулы, связывающей эти переменные, или соответствующего знания эксперта. Минимальный набор правил может быть построен автоматически с использованием нейронной сети.

## 2.6. Проектирование базы правил машины вывода

При решении большинства задач проектирования встроенных интеллектуальных систем поддержки принятия решений на основе умных сенсоров информацию, необходимую для их решения, можно разделить на две части: численную (количественную), полученную с физических сенсоров, и лингвистическую (качественную), поступающую от эксперта. Наиболее простым способом является использования лингвистической информации, поступающей от эксперта, в виде нечетких продукционных правил с заданными или выбранными из некоторого набора функциями принадлежности.

В случае, когда возникает необходимость спроектировать интеллектуальные системы поддержки принятия решений для БСС, но в наличии имеется только численные данные, мы сталкиваемся с серьезными проблемами.

Одним из путей их разрешения являются нейро-нечеткие системы типа *ANFIS* (адаптивная нейро-нечеткая система вывода). Они обладают многими достоинствами, однако сдерживающим моментом является длительность наполнения их знаниями в процессе итеративного обучения. К тому же, объем кода *ANFIS* пока превышает объем памяти сенсорного узла *Smart Sensors*. Поэтому возникает необходимость наличия вершины-координатора с достаточно мощными вычислительными возможностями, которая после обучения передает сформированные базы правил заинтересованным вершинам.

Более удобным вариантом являются самообучающиеся нейро-нечеткие системы типа нечетких карт Кохонена. Они не требуют создания обучающих выборок и могут обучаться в процессе функционирования сети. Кроме того, возникает возможность создания распределенной карты Кохонена на узлах сенсорной сети по принципу «один сенсор - один нейрон». В этом случае обучение и создание базы правил для одного или нескольких узлов может происходить без участия вершины-координатора.

Но наиболее простым и универсальным способом создания базы правил является разбиение пространств входных и выходных параметров на гранулы и построение базы правил путем установления нечеткого соответствия между обучающей выборкой и парами входных и выходных гранул. Достоинством этого простого и эффективного метода является возможность объединения численной информации, представленной в виде обучающих данных, с лингвистической информацией, имеющей вид базы правил, за счет пополнения имеющейся базы правилами, созданными на основе численных данных.

### 3. Реализация узла Smart Sensors

#### 3.1. Общая схема узла Smart Sensors

Общая схема узла *Smart Sensors* (для краткости – нечеткого сенсора) представлена на рис. 2.

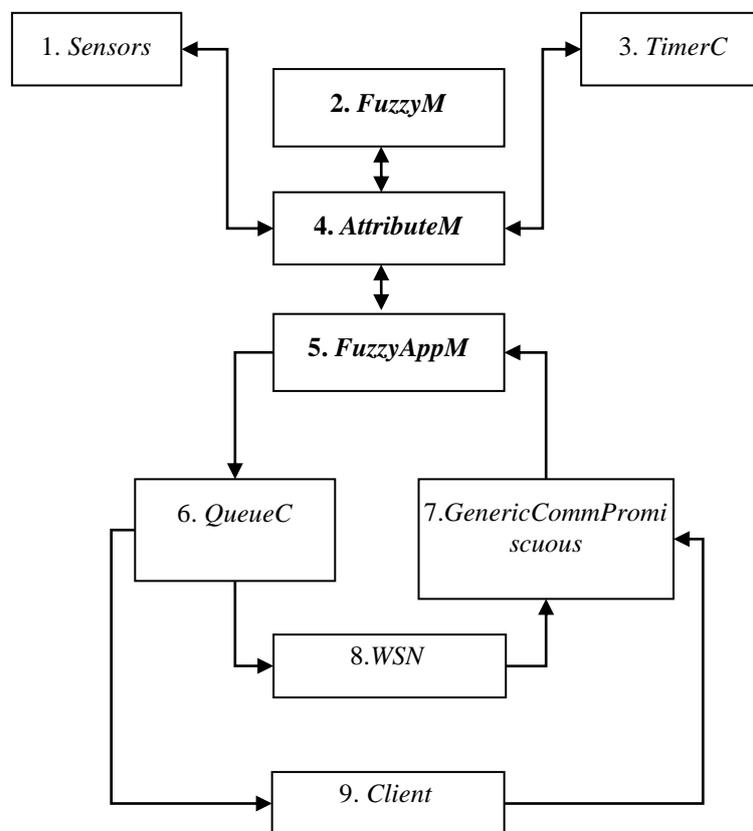


Рис. 2. Общая схема нечеткого узла *Smart Sensors* (нечеткий сенсор  $\Omega$ )

Здесь: 1. Физические датчики. 2. Машина нечеткого вывода. 3. Таймер. 4. Блок хранения базы правил, списков внутренних и внешних атрибутов и их значений. 5. Блок первичной обработки запросов. 6. Очередь исходящих запросов и сообщений. 7. Блок проверки входящих запросов и сообщений. 8. Интерфейс с сенсорной сетью. 9. Пользовательский интерфейс.

#### 3.2. Основные определения

В дальнейшем будем пользоваться следующей терминологией. Пусть задан сенсорный узел  $\Omega$ . Тогда:

- Физический сенсор – измерительное устройство, установленное на узле  $\Omega$ .
- Физический атрибут внутренний – значение, полученное физическим сенсором, установленном на данном узле  $\Omega$  (влажность, температура и т.д.). В сенсорном узле  $\Omega$  хранятся списки имен и значений внутренних физических атрибутов.

- Физический атрибут внешний – значение, полученное узлом  $\Omega$  от другого узла БСС. В сенсорном узле  $\Omega$  хранятся списки имен и значений внешних физических атрибутов.
- Виртуальный сенсор – вложенное в данный узел  $\Omega$  устройство вычисления (машина вывода) для значения величины, которая не может быть непосредственно измерена, но является функцией других известных величин (например, комфортность, престижность, опасность и т.д.).
- Виртуальный атрибут внутренний (выходной) – значение величины, вычисленное виртуальным сенсором  $\Omega$ . В сенсорном узле  $\Omega$  хранятся списки имен, и значений внутренних виртуальных атрибутов и баз правил для их вычисления.
- Виртуальный атрибут внешний (входной) – значение величины, полученное  $\Omega$  от другого узла. В сенсорном узле  $\Omega$  хранятся списки имен и значений внешних виртуальных атрибутов.
- База правил виртуального атрибута – нечеткие правила для вычисления виртуального атрибута с необходимыми функциями принадлежности.
- (Нечеткая) машина вывода – нечеткий контроллер для вычисления численных значений виртуальных атрибутов.

### 3.3. Основные функции узла в *Smart Sensors*

Данная система является многофункциональной и предназначена для обработки различных запросов пользователя, настройки и управления сетью. Система позволяет измерять различные параметры окружающей среды и использовать их для вычисления атрибутов, которые являются характеристиками объекта, среды и т.п. Система является объединением таких компонентов как пользователь (компьютер) и беспроводная сенсорная сеть (БСС), один из узлов которой (нечеткий сенсор) представлен на рис. 2. Пользователь задает характеристики объекта (виртуальные атрибуты), значения которых надо вычислить, и правила, при помощи которых эти атрибуты будут подсчитаны. Если узел не имеет достаточно данных для вычисления атрибутов, он может запросить их у другого узла.

Нечеткий сенсорный узел в *Smart Sensors* выполняет следующие функции:

- снятие показаний датчиков и сохранение значений в список значений внутренних физических атрибутов;
- получение значений физических атрибутов от других узлов и сохранение их в список значений внутренних и внешних физических атрибутов;
- получение запроса от пользователя или другого узла на предоставление списков значений имеющихся в  $\Omega$  физических атрибутов и отправление пользователю этих списков;
- получение значений виртуальных атрибутов от пользователя или других узлов и запоминание их в список значений внешних виртуальных атрибутов;
- вычисление значений внутренних виртуальных атрибутов узла  $\Omega$  с помощью машины вывода, используя списки внутренних и внешних значений физических атрибутов, и запоминание их в список внутренних виртуальных атрибутов;
- вычисление значений внутренних виртуальных атрибутов с помощью машины вывода, используя списки значений внутренних и внешних физических атрибутов, списки значений внутренних и внешних виртуальных атрибутов и запоминание их в список значений внутренних виртуальных атрибутов;
- отправление по запросу пользователю или другому узлу списков внутренних и внешних значений от пользователя и других узлов;
- пополнение списков имен внутренних виртуальных атрибутов и базы правил узла  $\Omega$  на основе полученных от пользователя или других узлов списка новых виртуальных атрибутов и правил для их вычисления;
- удаление на основе команды пользователя внутренних виртуальных атрибутов и их значений из списков, а также удаление из базы правил для вычисления удаленных атрибутов;
- пересылка по запросу пользователя или других узлов базы правил для вычисления виртуальных атрибутов.

### 3.4. Заполнение базы правил узла *Smart Sensors*

Первоначально каждый узел  $\Omega$  имеет некоторый список физических атрибутов, который соответствует его набору датчиков, и пустую оболочку машины вывода (2) и пустой список виртуальных атрибутов (4). Пользователь может добавить в список виртуальных атрибутов узла  $\Omega$  виртуальные внутренние (выходные) атрибуты и послать базу правил виртуального атрибута в машину вывода (2). Если в базе правил виртуального атрибута содержатся новые внешние физические и виртуальные атрибуты, то вносятся необходимые изменения в списки внешних физических и виртуальных атрибутов (4) узла  $\Omega$  и указываются имена узлов сети (8), от которых эти атрибуты могут быть получены и маршруты для их получения. В этом случае возникает иерархическая структура узлов сети (8), снабжающих узел  $\Omega$  необходимой информацией.

После установления базы правил виртуального атрибута в машину вывода в узел  $\Omega$  (нечеткий сенсор) начинает измерять значения внутренних физических атрибутов (1), получать от сети (8) и пользователя (9) значения внешних физических и виртуальных атрибутов. Машина вывода (2) производит вычисления внутренних виртуальных атрибутов. Полученные значения внутренних физических и виртуальных атрибутов сохраняются в соответствующих списках (4) и по требованию предоставляются пользователю (9) или другим узлам сети (8).

При необходимости пользователь (9) или другие узлы сети (8) могут добавлять в узел  $\Omega$  или исключать из него виртуальные атрибуты и их из базы правил. При этом должны проводиться необходимые изменения топологии сети (8).

### 3.5. Метаправила

По аналогии с экспертными системами, метаправилом также будем называть группу правил, отвечающей за использования машиной вывода в данный момент времени той или иной группы правил. Как и ЭС, в нечеткой базе знаний не существует способа, который позволял бы одному правилу вызывать другое, как если бы правила были процедурами. Метаправила, таким образом, существенно отличаются от обычных правил, поскольку они направляют ход измерений и вычислений атрибутов, а не принимают непосредственное участие в процессе их вычисления. Для управления передачей баз правил по сети (для управления знаниями) могут также использоваться нечеткие производные правила. Мы будем называть их метаправилами. Таким образом, возникает иерархия из двух типов нечетких сенсоров, верхним уровнем которых являются метасенсоры, или сенсоры-координаторы, а нижним – обычные нечеткие сенсоры, организованные в тематические блоки, соответствующие, например, уровням представления данных, перечисленным во введении. Безусловно, их роли будут меняться в зависимости от того, какие блоки правил загружены в них в настоящий момент.

В разделе 4 приводятся несколько примеров использования *Smart Sensors* для интеллектуальной поддержки решений в БСС в различных прикладных областях.

## 4. Области использования *Smart Sensors*

### 4.1. Использование *Smart Sensors* для управления БСС

При использовании знаний при управлении БСС возникает некоторая аналогия с идеологией активных сетей, в которых каждый пакет данных сопровождается капсулами с правилами для их обработки сервером (в нашем случае – узлом). В нашем случае действительно данные сопровождаются некоторыми правилами для их обработки, но предполагается, что существуют два вида уровня трафика и трафик знаний, причем трафик знаний ставит целью минимизацию трафика знаний. В активных сетях автономного трафика знаний не существует. В нашей системе трафика знаний используется специализированный язык представления знаний (в нашем случае – нечеткий язык представления знаний).

С помощью нечетких сенсоров могут быть реализованы:

- алгоритмы маршрутизации;
- оптимальное управление расходом энергии в сети;

- управление трафиком сообщений в сенсорной сети (активные сети);
- управление качеством обслуживания.

Приведем два примера правил вычисления внутренних виртуальных атрибутов узла для соответствующих баз правил.

ПРИМЕР 1. (вычисление виртуального атрибута качества обслуживания).

$R^{(1)}$ : **ЕСЛИ** (среднее использование каналов – НИЗКОЕ И средний процент запаздывания в канале – ВЫСОКИЙ) **ТО** (возможность возникновения очередей – СРЕДНЯЯ).

ПРИМЕР 2. (вычисление виртуального атрибута качества маршрута для занесения его в список возможных маршрутов).

$R^{(1)}$ : **ЕСЛИ** ( $LS=$ ВЫСОКАЯ И  $LC=$ ВЫСОКАЯ И  $NE=$ ВЫСОКАЯ И  $NH=$ НИЗКАЯ) **ТО** (запомнить в памяти – ОЧЕНЬ ВОЗМОЖНО)

$R^{(2)}$ : **ЕСЛИ** ( $LS=$ НИЗКАЯ И  $LC=$ ВЫСОКАЯ И  $NE=$ НИЗКАЯ И  $NH=$ ВЫСОКАЯ) **ТО** (запомнить в памяти – МАЛО ВОЗМОЖНО).

Здесь  $LS$  – надежность канала,  $LC$  – пропускная способность канала,  $NE$  – заряд узлов,  $NH$  – число узлов.

Необходимость использования в этой задаче нечетких правил связана с тем, что различные параметры сети или маршрутов описываются в разных метриках и могут быть сведены в один критерий только на лингвистическом уровне.

## 4.2. Использование *Smart Sensors* для создания нечетких баз в БСС

При использовании *TinyOS* и *TinyDB* в БСС возникает возможность создания нечеткой *TinyDB* и нечеткой активной *TinyDB*. Такая возможность существует и достаточно легко реализуется, но не представляется целесообразной, так как запросы к сенсорной сети и представление нечетких модификаторов с помощью *TinyDB* ограничены по возможностям, а *TinyDB* занимает большой объем памяти. Поэтому использование для обработки нечетких запросов расширенного на нечеткий случай *SQL* было бы неэффективно.

Поэтому в *Smart Sensors* на основе нечетких сенсоров осуществлена замена функций обычной реляционной БД для сенсорных сетей (*TinyDB*) и разработан специальный нечеткий язык запросов и коммуникаций в БСС (*FuzzyDB*), который может использоваться:

- для запросов к сенсорной сети (представляя как нечеткость запроса, так и нечеткость самих данных сети);
- для создания нечетких триггеров;
- для создания нечетких активных баз данных;
- для общения нечетких сенсоров между собой.

## 4.3. Использование *Smart Sensors* для агрегации данных

В настоящее время в БСС используются пять общих, проблемно-ориентированных методов слияния данных: ассоциация данных, оценка данных (уровень 1), слияние данных для определения идентичности, распознавание образов (уровень 2), и искусственный интеллект (уровень 3). Каждый метод соответствует нескольким специальным технологиям, ассоциация данных – критерию качества и технике селекции, оценка данных – Калмановской фильтрации, слияние данных для определения идентичности – байесовскому принятию решений, правдоподобным рассуждениям Демпстера-Шейфера и адаптивным нейронным сетям, распознавание образов – методам кластеризации, и, наконец, искусственный интеллект – экспертным системам, архитектуре доски объявлений и нечеткой логике.

Технологии слияния данных, сгруппированные по уровням слияния, различаются по уровню сложности информации, которую они предоставляют. В существующих до настоящего моментах системах слияния данных от многих сенсоров внимание обычно сосредоточено на извлечении ин-

формации низкого уровня. В нашей системе мы рассматриваем информацию более высокого уровня, т.е. информацию, связанную со многими обычными физическими величинами, представленными в неаналитической форме. В нашей системе в этих сложных случаях мы используем нечеткие сенсоры, которые вычисляют и передают лингвистические значения результатов агрегации и слияния численных измерений, причем могут быть используются два метода для агрегации базовых измерений. Первый из них осуществляет комбинацию существенных результатов с помощью выраженных в виде правил связей между ними. Во втором методе агрегация и слияние данных реализована с помощью нечеткого разбиения числового многомерного пространства измерений. Наилучший результат дает гибридизация этих двух подходов.

Однако нечеткие сенсоры могут быть использованы практически и на более низких уровнях слияния данных, например, для фильтрации данных, для выделения атрибутов, для распознавания образов. Алгоритмы слияния данных, записанные в продукционной форме, легко декомпозируются, им присущ иерархический характер. Поэтому слияние данных внутри сети может быть осуществлено иерархией узлов, между которыми динамически распределяются знания, необходимые для обработки данных в конкретном узле.

Пример 3 иллюстрирует механизм слияния данных с помощью нечетких сенсоров для построения виртуальных сенсоров, измеряющих комфортность аудитории университета.

**ПРИМЕР 3** (вычисление значения виртуального атрибута комфортности комнаты на основе температуры, влажности и освещенности)

$R^{(1)}$ : **ЕСЛИ** (температура – СРЕДНЯЯ И влажность – ВЫСОКАЯ И освещенность – ВЫСОКАЯ), **ТО** (комфортность ОЧЕНЬ ВЫСОКАЯ).

$R^{(2)}$ : **ЕСЛИ** (температура – НИЗКАЯ И влажность – ВЫСОКАЯ И освещенность – НИЗКАЯ), **ТО** (комфортность – НИЗКАЯ).

На рис. 3 представлена диаграмма передачи данных и знаний по сети при обработке запроса «Укажите наиболее комфортную из аудиторий?».

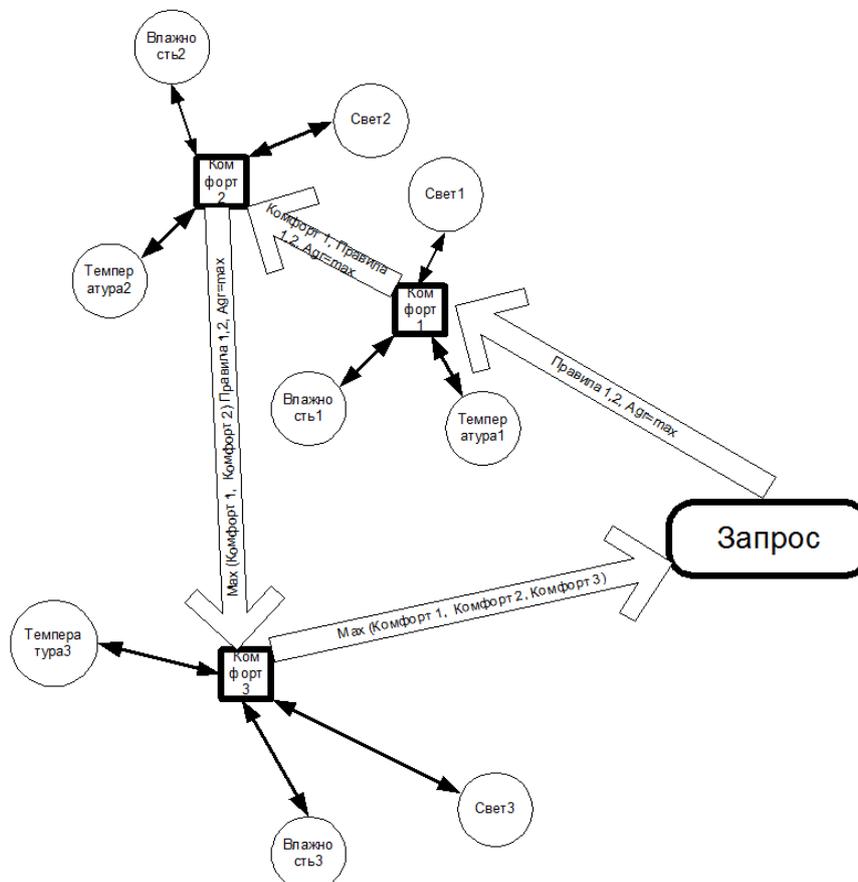


Рис. 3. Запрос «Укажите наиболее комфортную из аудиторий?»

В примере 4 процедура слияния данных используется при управлении частотой срабатывания противопожарных сенсоров в зависимости и температуры, измеряемой соседними датчиками.

ПРИМЕР 4. (Изменение скорости работы пожарного датчика температуры в зависимости от показаний соседних датчиков).

$R^{(1)}$ : ЕСЛИ (температура  $S_0$  – НИЗКАЯ И температура  $S_1$  – ВЫСОКАЯ И температура  $S_2$  – ВЫСОКАЯ И температура  $S_3$  – СРЕДНЯЯ), ТО (скорость работы  $S_0$  – ВЫСОКАЯ).

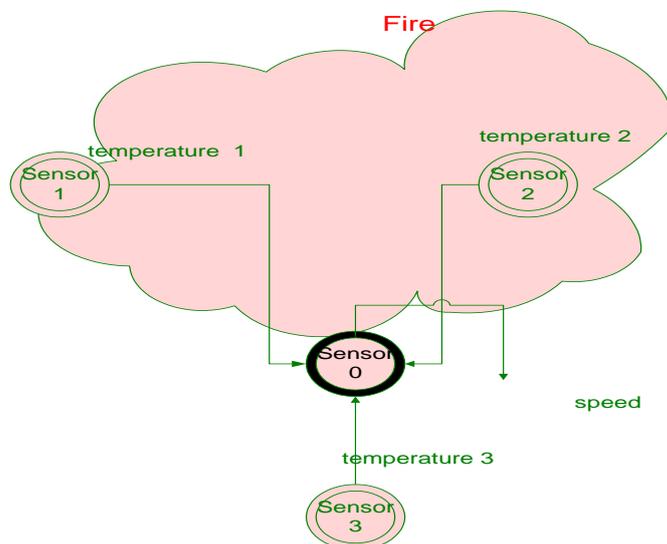


Рис.4. Изменение скорости работы противопожарного сенсора температуры  $S_0$  в зависимости от показаний 3-х соседних сенсоров  $S_1$ ,  $S_2$  и  $S_3$ .

#### 4.4. Использование **Smart Sensors** для создания нечетких распределенных ЭС

Беспроводные сенсорные сети (БСС) представляют собой громадную динамическую базу данных, однако для эффективного использования информации необходима ее эффективная организация. Особенный интерес представляют подходы, позволяющие использовать БСС, как распределенную вычислительную среду интеллектуальной обработки данных и как хранилище этих методов, а не просто средство для их сбора и пересылки. Такие подходы должны также обеспечивать распределенное накопление, передачу, хранение и использование этих знаний. Один из подходов состоит в использовании экспертных систем, база знаний которых распределена по узлам БСС, и данные обрабатываются в этих узлах при использовании своей базы знаний и баз знаний соседних узлов.

Однако основная проблема заключается в том, что доставка данных в узел, в котором осуществляется слияние данных, само по себе может оказаться непростой и дорогостоящей проблемой с точки зрения расхода ресурсов, так как этот узел обычно считается заданным и максимально приближенным к пользователю. Поэтому назначение сенсорных узлов машинами для слияния данных должно иметь динамический характер, положение которых оптимизируется в зависимости от запроса, состояния сети или внешней среды. Вместе с изменением функции узла меняется и содержащаяся в нем база знаний.

Пример 5. При передаче функции центра слияния данных внутри кластера от вершины  $\Omega_1$  к вершине  $\Omega_2$  база знаний, реализующая эту функцию, также перемещается из вершины  $\Omega_1$  в вершину  $\Omega_2$ .

Пример 6. Пользователь запрашивает у вершины  $\Omega_1$  атрибут, который она не умеет вычислять. Тогда база знаний, необходимая для вычисления этого атрибута, должна быть послана в  $\Omega_1$  из ближайшего к нему узла, в котором эта база содержится.

Пример 7. При мониторинге динамического объекта при движении или другом изменении параметров события (цели) расположение этих узлов может меняться, а алгоритмы слияния данных – передаваться из старых узлов в новые, ближайшие к цели, а также удаляться из старых узлов.

Если в узле не хватает каких-либо входных атрибутов для выполнения запроса, то он обращается к окружающим узлам с просьбой о пересылке необходимых атрибутов или правил для их вычисления. Таким образом, предлагаемый подход заключается в максимальном приближении узлов слияния данных к событию, в создании иерархии обработки данных с вершиной в этом узле, и в возможности многоэтапной обработки запроса в узле (сначала узел получает или вычисляет необходимые для обработки запроса входные атрибуты, а потом вычисляет на их основе ответ на запрос пользователя). Процесс обработки запроса, выходя из узла обработки запроса, распределяется по сети и во времени, и снова возвращается в исходный узел. Универсальность представления этих алгоритмов в виде продукционных правил (нечетких сенсоров) позволяет максимально упростить эту процедуру за счет передачи не самих алгоритмов (программного кода), а их параметров (знаний). Для передачи нечетких правил от одного узла к другому должен использоваться специальный нечеткий язык представления знаний *FUSL (Fuzzy sensors language)*, аналогичный, например, языку *Full*, который используется корпорацией *STMicroelectronics* для передачи нечетких правил от нейро-нечеткой сети *AFM* контроллерам семейства *WARP*.

Таким образом, нечеткая распределенная база знаний для обработки запроса к распределенной базе данных обладает следующими свойствами:

- Функционирует, как распределенная экспертная система.
- Знания в продукционной форме могут передаваться между узлами.
- База знаний, нужная для обработки запроса, может быть послана в нужные узлы вместе с запросом.
- Для передачи нечетких знаний от узла к узлу используются специальный язык.

Программа обработки запроса максимально использует возможность параллельных вычислений и пишется не для отдельного узла, а для всей сети в целом.

#### 4.5. *Smart Sensors* как вложенные многоагентные системы

Высокий уровень интеллектуальности узлов-координаторов позволит использовать для их создания парадигму интеллектуальных агентов (путем добавления в них модели внешней среды, возможности самостоятельного целеполагания и планирования) что облегчит интеграцию симулятора БСС на базе ВМЭС в системы, использующие технологии многоагентных систем. Например, БСС с узлами, наделенные функциями интеллектуальных агентов, могут быть использованы в моделях боевых действий, мониторинга и управления гибкими производственными процессами, транспортными перевозками, спортивными играми.

Нечеткие сенсорные узлы могут обладать активностью, мобильностью, и коллективным поведением. Например, парадигма агентов может быть использована в следующих задачах БСС:

- Миграция приложений (наборов правил) для экономии памяти и энергии, например, передача функции головы кластера от одного сенсорного узла к другому в связи с истощением источника энергии у первоначального узла.
- Использование нечеткого сенсора в качестве мобильного агента, обходящего сенсорные узлы по кольцевому маршруту для сбора и обработки данных.
- Введение в сенсорный узел функции полезности, в соответствии с которой они, в зависимости от величины заряда своей батареи, соглашаются или отказываются быть посредником для передачи данных.
- Взаимодействие сенсоров - агентов для слияния данных (например, использование двух сенсоров для пеленгации объекта).

Приведем два примера метаправил, управляющих передачей приложений (базы знаний) от одного нечеткого сенсора к другому.

Пример 8 (метаправило, управляющее передачей приложений, вызванной мобильностью точки наблюдения).

Рассмотрим случай, когда передача параметров правил (приложения) от узла **S1** к узлу **S2** происходит в связи с ограниченностью энергетических ресурсов **S1**. Путь набор правил задает функцию головы кластера. Получаем правило

$R^{(1)}$ : **ЕСЛИ** (заряд узла  $S_1$  – НИЗКИЙ **И** узел  $S_2$  – БЛИЗКО к узлу  $S_1$  **И** заряд узла  $S_2$  – ВЫСОКИЙ), **ТО** (передать приложение узлу  $S_2$  – ВОЗМОЖНО).

Основной причиной миграции кодов приложений между вершинами в проводных и беспроводных сетях является оптимизации расхода энергии. Показывается, что поскольку БСС все равно являются гибкими и динамичными и сильно ограничены по памяти и питанию, миграция кодов приложений выгоднее для экономии питания и памяти, чем их инсталляция на все возможные вершины.

Пример 9 (метаправило для передачи параметров приложения в связи с мобильностью объекта мониторинга).

$R^{(1)}$ : Если (расстояние от узла  $S_1$  до цели ДАЛЕКО **И** расстояние от узла  $S_2$  до цели БЛИЗКО **И** наличие приложения у узла  $S_2$  невозможно), **ТО** (передать приложение узлу  $S_2$  НАВЕРНЯКА).

В БСС с обычными сенсорными узлами функция объем передачи знаний при мониторинге мобильного объекта минимален, так как передается только индикатор активности вершины. Вершины, которые видят объект, становятся активными, а которые перестают видеть – пассивными. В нашем случае вместе переключением вершины в активное состояние ей передается также и актуальный набор запросов по объекту мониторинга.

## Заключение. Перспективы развития Smart Sensors

Несмотря на простоту представления знаний в виде нечетких продукций, процесс задания функций принадлежности нечетких лингвистических переменных, извлечения правил и, особенно, их последующей отладки, представляет собой достаточно длительный процесс. Наличие обучающей выборки, отражающей мнение эксперта об эталонном процессе измерения, может значительно ускорить этот процесс. Поэтому ближайшей целью разработчиков *Smart Sensors* является использование, помимо нечетких сенсоров, нейро-нечетких сетей (вложенных в один узел или распределенных по БСС), генетических алгоритмов (вложенных в один узел или распределенных по сети) и искусственных иммунных систем (распределенных по сети). Они смогут управлять адаптацией и обучением нечетких сенсоров для оптимизации таких иерархических процессов в БСС, как кластеризация, фильтрация, агрегация, ассоциация, слияние данных. Это существенно повысит эффективность *Smart Sensors* в решении для таких задач, как слежение за стационарными и мобильными объектами, мониторинг внешней среды, зданий и производственных процессов. Кроме того, они могут управлять основными процессами в *Smart Sensors* – передачей данных и знаний по сети, оптимизацией расхода энергии, защитой сети от атак. При использовании модели гетерогенных БСС функции нейро-нечетких сенсоров будет целесообразно передать выделенным узлам с постоянными источниками питания и другими типами связи.

## Список литературы

1. C. Siva Ram Murthy, B.S. Manoj Ad Hoc Wireless Networks: Architectures and Protocols, Published by Prentice Hall PTR, Series: Prentice Hall Communications Engineering and Emerging Technologies Series, 2004. — P. 857.
2. Аверкин А.Н., Лавров Г.К. Создание универсального интерфейса высокого уровня для умного узла беспроводной сенсорной сети // Системный анализ в науке и образовании: сетевое научное издание. – Дубна, 2015. – №2. – [Электронный ресурс]. URL: <http://sanse.ru/download/239>.
3. Ярушкина Н. Г. Нечеткие нейронные сети // Новости искусственного интеллекта. — 2001. — № 2-3.
4. Батыршин И.З. Основные операции нечеткой логики и их обобщения. — Казань: Отечество, 2001. — С. 53-56.
5. Zadeh, Lotfi A. Fuzzy Logic, Neural Networks, and Soft Computing // Communication of the ACM, March 1994. — Vol. 37. — №. 3. — Pp. 77-84.
6. Святкина М.Н. Беспроводные сенсорные сети в интеллектуальных средах // Том. 2 Проблемы фундаментальной науки. Стратегические информационные технологии. Материалы Научной сессии НИЯУ МИФИ-2012 в 3 т., 1-4 февраля 2012. — С. 388.