

На правах рукописи



**Литвинов Владимир Геннадьевич**

**РАЗРАБОТКА И ПРИМЕНЕНИЕ ТИПОВЫХ РЕШЕНИЙ ДЛЯ  
РАСПАРАЛЛЕЛИВАНИЯ АЛГОРИТМОВ ЧИСЛЕННОГО  
МОДЕЛИРОВАНИЯ**

05.13.18 — Математическое моделирование, численные методы  
и комплексы программ

**АВТОРЕФЕРАТ**  
диссертации на соискание учёной степени  
кандидата технических наук

Самара — 2015

Работа выполнена в федеральном государственном автономном образовательном учреждении высшего образования «Самарский государственный аэрокосмический университет имени академика С.П. Королева (национальный исследовательский университет)» на кафедре информационных систем и технологий.

**Научный руководитель:**

доктор технических наук, доцент Востокин Сергей Владимирович

**Официальные оппоненты:**

Орлов Сергей Павлович, доктор технических наук, профессор, федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Самарский государственный технический университет», заведующий кафедрой «Вычислительная техника»;

Тарасов Вениамин Николаевич, доктор технических наук, профессор, федеральное государственное образовательное бюджетное учреждение высшего профессионального образования «Поволжский государственный университет телекоммуникаций и информатики», заведующий кафедрой «Программное обеспечение и управление в технических системах».

**Ведущая организация:**

Федеральное государственное бюджетное учреждение науки «Институт проблем управления сложными системами РАН», г. Самара.

Защита состоится 6 ноября 2015 г. в 10 часов на заседании диссертационного совета Д212.215.05 ФГАОУ ВО «Самарский государственный аэрокосмический университет имени академика С.П. Королева (национальный исследовательский университет)», по адресу: 443086 Россия, г. Самара, Московское шоссе, д. 34.

С диссертацией можно ознакомиться в библиотеке и на сайте ФГАОУ ВО «Самарский государственный аэрокосмический университет имени академика С.П. Королева (национальный исследовательский университет)», [http://www.ssau.ru/files/resources/dis\\_protection/diss\\_Litvinov\\_V\\_G.pdf](http://www.ssau.ru/files/resources/dis_protection/diss_Litvinov_V_G.pdf).

Автореферат разослан 14 сентября 2015 г.

Ученый секретарь  
диссертационного  
совета, д.т.н., доцент



С. В. Востокин

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность.** Роль высокопроизводительной вычислительной техники в области математического моделирования чрезвычайно важна. Её применение позволяет сократить время решения задач моделирования, повысить размерность и точность решаемых задач, проводить исследования, которые принципиально невозможны на последовательной ЭВМ.

Переход от последовательного численного метода к параллельному и, далее, к вычислительным экспериментам на высокопроизводительных ЭВМ является самостоятельной научной и инженерной проблемой. Она обусловлена тем, что к сложности исходного численного метода добавляется сложность организации коммуникации, синхронизации, распределения вычислений по процессорам. Данная проблема выделена академиком Г.И. Марчуком как фундаментальное научное направление, называемое «проблемой отображения».

Одним из подходов к решению «проблемы отображения» является использование типовых решений (шаблонов, паттернов, алгоритмических скелетов) для распараллеливания численных методов разного назначения. Паттерны впервые описаны в работе Э. Гамма, Р. Хелма, Р. Джонсона и Дж. Влиссидеса для объектно-ориентированного проектирования. В области типовых решений для параллельных научных вычислений известны работы отечественных и зарубежных ученых: М.И. Коула, С. Макдональда, П.К. Берзигиярова, В.Э. Малышкина, С.В. Востокина и других. В них представлено около двух десятков различных типовых решений в области параллельного программирования.

Однако актуальной задачей исследований остаётся разработка общих подходов для описания разных типовых решений. В диссертации предлагается и исследуется общая модель вычислительного процесса, позволяющая строить специализированные модели типовых решений. Используется графическая форма представления модели типового решения, удобная для конструирования параллельных численных алгоритмов и программ.

В качестве важного приложения методов синтеза параллельных алгоритмов на базе типовых решений в диссертации исследованы методы построения интеллектуальных систем на базе нейронных сетей, предложенных С. Осовским, С. Хайкиным, А.Н. Горбанём, с целью разработки эффективных параллельных программ обучения гиперрадиальных нейронных сетей для решения задач прогнозирования.

Процесс разработки параллельных программ в настоящее время отличается относительно низкой автоматизацией. Поэтому большое практическое значение имеют снижение стоимости и сокращение сроков проведения вычислительных экспериментов на высокопроизводительных ЭВМ. В этой связи в диссертации рассматривается проблема построения программного комплекса на основе библиотеки типовых решений и актуальное направ-

ление интеграции программного комплекса в инфраструктуру организации научных вычислений на основе web-сервиса и в современные процессы разработки.

**Цель и задачи исследований.** Целью исследований является разработка методики построения эффективных параллельных алгоритмов и комплексов программ на основе математической модели вычислительного процесса; применение данного подхода при постановке численных экспериментов на высокопроизводительных системах в задаче обучения гиперрадиальной нейронной сети.

Задачи исследований, соответствующие поставленной цели диссертационной работы.

- а) Анализ актуальных проблем разработки программных комплексов высокопроизводительных научных вычислений.
- б) Разработка общей математической модели вычислительного процесса для описания различных типовых решений в области высокопроизводительных научных вычислений.
- в) Разработка графической модели алгоритма для типового решения «портфель задач» на основе математической модели вычислительного процесса и графической нотации Templet.
- г) Построение методики синтеза параллельного алгоритма на примере алгоритма обучения гиперрадиальной нейронной сети с использованием модели типового решения «портфель задач».
- д) Разработка программного комплекса численного моделирования на основе типового решения «портфель задач» для использования на высокопроизводительных вычислительных системах.
- е) Проведение вычислительных экспериментов для анализа эффективности распараллеливания и сложности разработки с использованием программного комплекса на основе типового решения «портфель задач».

**Научная новизна диссертационной работы.** В диссертации получены следующие новые научные результаты.

- а) Построена новая математическая модель вычислительного процесса для описания различных типовых решений. В отличие от традиционных моделей типовых решений, она строго формализована с использованием аппарата темпоральной логики Лампорта, является основой для кроссплатформенной эффективной реализации параллельных алгоритмов.
- б) Разработана графическая модель алгоритма типового решения «портфель задач». В отличие от известных методов описания принципа организации вычислений в типовом решении, новая графическая модель алгоритма позволяет детально передать логику его функционирования без привязки к языку реализации.
- в) Предложена и описана на примере построения параллельного алго-

ритма обучения нейронной сети гиперрадиального типа новая методика синтеза параллельного алгоритма. Методика основана на нисходящей декомпозиции исходного последовательного алгоритма на основе графической модели параллельного алгоритма. В ней, в отличие от традиционных, не используются примитивы синхронизации и коммуникации языка и среды исполнения.

- г) Разработан программный комплекс и приведены результаты его экспериментального исследования на примере решения задачи обучения гиперрадиальной нейронной сети. В отличие от известных программных решений, использование общей математической модели типового решения позволило добиться кроссплатформенности программы с сохранением высокой эффективности её выполнения в разных средах: Windows, POSIX, MPI.

**Теоретическая и практическая значимость работы.** Теоретическая значимость работы состоит в том, что в ней предложена методика, расширяющая известные методики, основанные на типовых решениях. Новыми элементами методики являются: математическая модель вычислительного процесса и графическое представление параллельных алгоритмов в типовых решениях. Математическая модель является основой кроссплатформенной реализации параллельного алгоритма. Использование графической модели в типовых решениях обобщает их описание на произвольный алгоритмический язык реализации и программную платформу и задаёт точную декомпозицию параллельного алгоритма на составляющие последовательные процедуры и обрабатываемые ими типы данных.

Практическая значимость работы состоит в создании программного комплекса, позволяющего автоматизировать постановку вычислительного эксперимента на высокопроизводительной ЭВМ. Его использование сокращает время проведения вычислительного эксперимента, исключает возможные ошибки в параллельной программе, расширяет круг специалистов, применяющих высокопроизводительную вычислительную технику для решения задач математического моделирования. Программный комплекс размещён в сети интернет как компонент web-сервиса научных вычислений Templet Web (<http://templet.ssau.ru>).

Прикладное значение также имеют реализованные в работе параллельный алгоритм и программа обучения и оптимизации структуры гиперрадиальной нейронной сети. Их применение позволяет снизить вычислительную сложность на стадии функционирования нейронной сети. В модельных экспериментах с нейронной сетью продемонстрирована эффективность предложенной в работе методики разработки параллельных алгоритмов.

**Соответствие паспорту специальности.** Работа соответствует следующим пунктам паспорта заявленной научной специальности: (п. 1) разработка новых математических методов моделирования объектов и явлений;

(п. 3) разработка, обоснование и тестирование эффективных вычислительных методов с применением современных компьютерных технологий; (п. 4) реализация эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента.

**Реализация результатов работы.** Диссертационная работа выполнена в рамках работ по проекту «Разработка комплекса технологий использования ресурсов суперкомпьютера «Сергей Королёв» в целях развития инновационной и научно-образовательной среды университета» (мероприятия 3.3 Программы развития национального исследовательского университета).

Результаты диссертационной работы внедрены в научной деятельности факультета летательных аппаратов СГАУ в 2013 году и применяются при выполнении НИР для анализа многомерных динамических систем и процессов с использованием суперкомпьютера «Сергей Королёв», о чем имеется акт внедрения.

**Методы исследований.** В работе использованы темпоральная логика Лампорта, графические методы описания алгоритмов, имитационное моделирование, теория искусственных нейронных сетей.

**Апробация работы.** Основные результаты работы были представлены на следующих всероссийских и международных конференциях: Всероссийской молодежной научной конференции с международным участием «X Королёвские чтения» (Самара, 2009); Международной конференции с элементами научной школы для молодежи «Перспективные информационные технологии для авиации и космоса (ПИТ-2010)» (Самара, 2010); Всероссийской молодежной научной конференции с международным участием «XI Королёвские чтения» (Самара, 2011); Шестой Международной конференции Института проблем управления имени В.А. Трапезникова РАН «Параллельные вычисления и задачи управления» (Москва, 2012); Научно-технической конференции с международным участием «Перспективные информационные технологии в научных исследованиях, проектировании и обучении (ПИТ-2012)» (Самара, 2012); Научно-технической конференции с международным участием «Перспективные информационные технологии (ПИТ 2013)» (Самара, 2013).

**Авторский вклад.** Все результаты, изложенные в диссертации, получены автором лично. В работах, выполненных совместно, автору принадлежат части, относящиеся к типовому решению «портфель задач» и алгоритм обучения нейронной сети. Автор лично осуществлял проведение численных экспериментов, обработку, анализ и интерпретацию полученных результатов.

**Достоверность результатов работы.** Достоверность полученных в диссертации решений подтверждена корректными математическими выкладками, результатами вычислительных экспериментов на высокопроизводительной технике, а также соответствием результатов математического

моделирования реальным процессам.

**Основные положения, выносимые на защиту.**

- а) Математическая модель вычислительного процесса для описания различных типовых решений на основе темпоральной логики Лампорта.
- б) Графическая модель алгоритма типового решения «портфель задач».
- в) Методика нисходящей декомпозиции последовательного алгоритма на основе графической модели алгоритма типового решения «портфель задач»; параллельный алгоритм обучения нейронной сети гиперрадиального типа, построенный на её основе.
- г) Программный комплекс численного моделирования, реализующий типовое решение «портфель задач»; результаты его функционирования на примере задачи обучения гиперрадиальной нейронной сети.

**Публикации по теме диссертации.** По теме диссертации опубликовано 12 работ, в том числе 3 статьи в журналах, рекомендованных ВАК РФ, одно свидетельство о регистрации программы для ЭВМ, 6 работ в материалах и трудах международных и всероссийских конференций, 2 тезиса доклада.

**Структура и объем работы.**

Диссертация состоит из введения, трех глав, заключения и трех приложений. Общий объем диссертации 120 страниц. Диссертация содержит 7 таблиц, 34 рисунка и список литературы из 130 наименований.

## СОДЕРЖАНИЕ РАБОТЫ

Диссертация состоит из введения, трех глав, заключения и списка использованной литературы.

**Во введении** обоснована актуальность темы диссертации, определена цель и сформулированы задачи работы, изложена постановка задачи диссертационной работы, научная новизна и практическая значимость полученных результатов. Приводятся основные положения, выносимые на защиту.

**В первой главе** приводится обзор актуальных проблем разработки параллельных программ и выполнено построение математической модели вычислительного процесса для описания различных типовых решений.

Для описания типовых решений предлагается рассматривать алгоритмы, информационно-логическую структуру которых можно описать ориентированным графом вида

$$G_M \triangleq (P, C), P \subset N, C \subset N \times N, \quad (1)$$

в котором вершины  $P$  представляют собой процессы, а дуги  $C$  каналы, по которым передаются сообщения между процессами. Для описания вычислительного процесса, порождаемого такими алгоритмами, применена темпоральная логика Лампорта. Состояние вычислительного процесса — это

значения множества переменных, связанных с процессами и каналами:

$$f \triangleq (i \in P : a_i; (i, j) \in C : e_{i,j}, d_{i,j}), \quad (2)$$

где  $a_i \in \{1, 0\}$ ,  $b_{i,j} \in \{1, 0\}$ ,  $d_{i,j} \in \{1, 0\}$ .

Логические переменные  $a_i$  обозначают пассивное 0 или активное 1 состояние процесса  $i$ ;  $e_{i,j}$  пассивное 0 или активное 1 состояние канала  $(i, j)$ ;  $d_{i,j}$  — направление передачи сообщений по двунаправленным каналам ( $1 - i \rightarrow j$ ;  $0 - j \rightarrow i$ ).

В начальном состоянии некоторые каналы находятся в активном состоянии с передачей сообщений в направлении  $i \rightarrow j$ , все процессы находятся в пассивном состоянии:

$$I \triangleq \forall i \in P : \neg a_i \wedge \forall (i, j) \in C : (e_{i,j} \wedge d_{i,j}) \vee (\neg e_{i,j} \wedge \neg d_{i,j}). \quad (3)$$

Действия, выполняемые каналами, описываются формулой

$$A_c(i, j) \triangleq \{e_{i,j} \wedge \neg e'_{i,j} \wedge ((d_{i,j} \wedge d'_{i,j}) \vee (\neg d_{i,j} \wedge \neg d'_{i,j}))\} \vee \\ \vee \{\neg e_{i,j} \wedge e'_{i,j} \wedge ((d_{i,j} \wedge \neg d'_{i,j}) \vee (\neg d_{i,j} \wedge d'_{i,j}))\}. \quad (4)$$

Действия описывают соседние в истории вычислительного процесса состояния, переменные следующего состояния обозначены апострофом. Согласно формуле (4) состояние канала периодически меняется с активного на пассивное, а направление передачи данных с прямого на обратное.

Действия, выполняемые процессами, описываются формулой

$$A_p(i) \triangleq \{\neg a_i \wedge a'_i \wedge (\exists! j : e_{i,j} \wedge \neg e'_{i,j} \wedge \neg d'_{i,j}) \wedge \neg (\exists j : e_{j,i} \wedge \neg e'_{j,i} \wedge d'_{j,i})\} \vee \\ \vee \{\neg a_i \wedge a'_i \wedge (\exists! j : e_{j,i} \wedge \neg e'_{j,i} \wedge d'_{j,i}) \wedge \neg (\exists j : e_{i,j} \wedge \neg e'_{i,j} \wedge \neg d'_{i,j})\} \vee \\ \vee \left\{ a_i \wedge \neg a'_i \wedge \bigvee_{k: (i,k) \in C} (\neg e_{i,k} \wedge e'_{i,k} \wedge d'_{i,k}) \right\} \vee \\ \vee \left\{ a_i \wedge \neg a'_i \wedge \bigvee_{k: (k,i) \in C} (\neg e_{k,i} \wedge e'_{k,i} \wedge d'_{k,i}) \right\} \vee \{a_i \wedge \neg a'_i\}. \quad (5)$$

Процесс становится активным, когда из смежного с ним канала поступает сообщение. При этом такой канал сам переходит в пассивное состояние. Среди нескольких каналов, готовых передать сообщение в процесс, выбирается произвольный канал. Когда процесс завершит работу и перейдет в пассивное состояние, по одному или нескольким каналам от него могут передаваться сообщения в другие процессы.

С учетом выражений (4) и (5), действия, выполняемые моделируемым вычислительным процессом, можно представить в виде

$$N \triangleq \bigvee_{(i,j) \in C} A_c(i, j) \vee \bigvee_{i \in P} A_p(i). \quad (6)$$

Способ планирования вычислительного процесса характеризуется выражением

$$F \triangleq (\forall i \in P : WF_f(a_i \wedge \neg a'_i)) \wedge (\forall (i, j) \in C : WF_f(e_{i,j} \wedge \neg e'_{i,j})), \quad (7)$$

подразумевающим справедливую в слабом смысле стратегию планирования.

Окончательно получаем модель вычислительных процессов для представления типовых решений:

$$\Phi_M \triangleq I \wedge \square [N]_f \wedge F. \quad (8)$$

По построению вычислительного процесса порядок смены состояний, удовлетворяющий формуле (8), легко воспроизвести дискретно-событийной имитационной моделью. Для этого организуется очередь каналов, находящихся в активном состоянии. На каждом шаге моделирования случайным образом выбирается активный канал. Далее он переводится в неактивное состояние, выполняется действие процесса и отправка сообщений из данного процесса.

Для представления типового решения далее требуется определить сообщения, передаваемые в каналах и методы процессов, запускаемые при их обработке. Для этого применена графическая нотация Templet. Модели процессов Templet напоминают блок-схемы алгоритмов рис. 1–3. Однако имеется несколько начальных вершин, соответствующих портам. Не делается различий между процедурой и принятием решения. Поток управления может пройти через вершину-метод, при этом метод может не выполняться, если не соблюдены некоторые условия. Данные свойства позволяют удобно описать выбор метода обработки поступающих в процесс сообщений. В отличие от блок-схем точно задан контекст запуска метода. Определено, какие сообщения пересылаются между процессами и их порядок. Это позволило провести анализ непротиворечивости графического описания алгоритма. Семантика исполнения модели по построению гарантирует отсутствие состояния состязания (race condition) при реализации алгоритма в общей памяти и состояния тупика (deadlock). Хотя для моделей вида (8) возможна остановка в незавершенном состоянии, данная ситуация более проста, чем состояние тупика.

Типовыми схемами решения многих переборных и оптимизационных задач являются схемы «управляющий — рабочие». Рассматривается модель одной из общих схем управления вычислениями из данного класса: «вычисление с портфелем задач». Это схема с активными рабочими, реализующая управление pull-типа. Здесь иллюстрируется описание более сложного протокола взаимодействия процессов с несколькими типами сообщений (по сравнению с исходными (4) протоколами «запрос-ответ»).

На рис. 1 показана композиция процессов схемы «портфель задач», соответствующая графу (1). Обработка заданий выполняется по инициативе

рабочих процессов  $w_1, \dots, w_N$ , что видно из направления каналов  $p \rightarrow p_1: Link, \dots, p \rightarrow p_N: Link$  от рабочих к управляющему процессу.

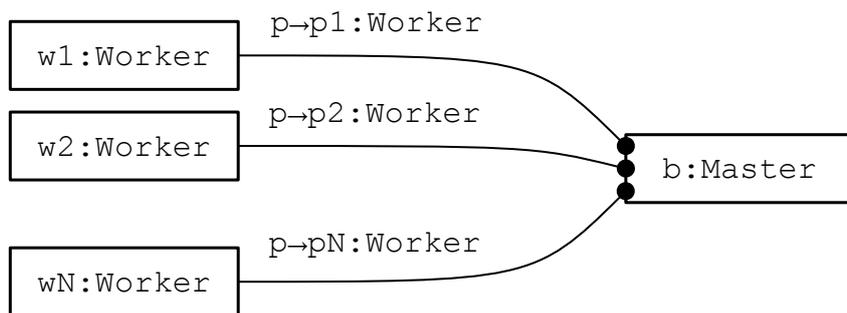


Рисунок 1 — Композиция процессов при вычислении с портфелем задач

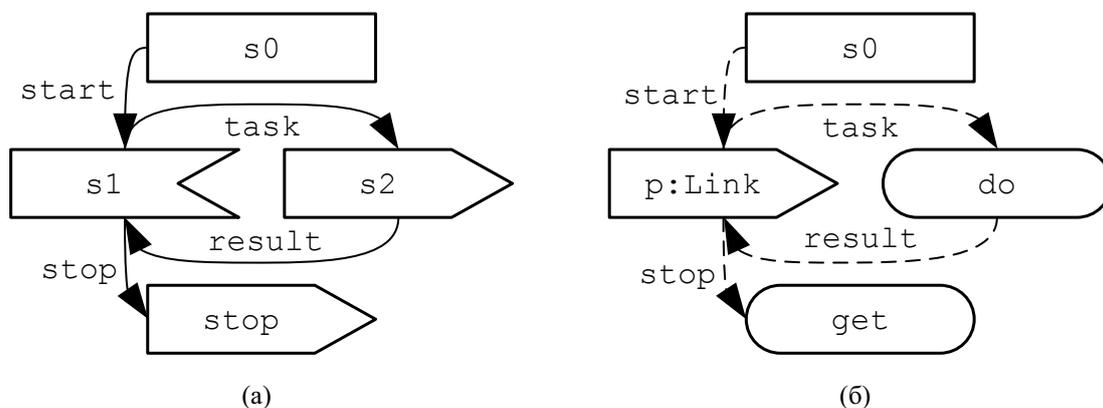


Рисунок 2 — Протокол канала (а) и рабочий процесс (б) алгоритма вычисления с портфелем задач

Роль рабочих процессов рис. 2(б) заключается в периодическом получении задания от управляющего процесса в сообщении *task* и отправки результата его обработки *result*. Обработка выполняется в методе *do*. Цикл обработки начинается по инициативе рабочего с отправки сообщения *start* и заканчивается при получении от управляющего процесса сообщения *stop*. Если потребуется, в методе *stop* можно выполнить операцию очистки.

На рис. 3 показан фрагмент управляющего алгоритма, реализованного в процессе *Master*, относящийся к порту  $p_N$ . Методы процесса *get*, *put*, *if\_task* и *start* реализуют специфические для решаемой задачи операции: *get* — извлечение очередной задачи; *put* — сохранение результата; *if\_task* — проверку наличия задачи для обработки. Метод *start* может использоваться для инициализации.

Методы *if\_active*, *wait\_N*, *stop\_N*, *waiting\_N* являются универсальными для произвольного численного метода. Метод *if\_active* возвращает истинное значение, если имеются активные рабочие процессы. То есть, имеются каналы, не находящиеся в состоянии «доставлено сообщение *start* или

*result*», что в программной реализации определяется простым подсчетом при поступлении и отправке сообщений. Метод  $wait_N$  устанавливает признак, что канал  $N$  находится в состоянии ожидания появления заданий, метод  $waiting_N$  возвращает значение данного признака. Метод  $stop_N$  используется при отправке сообщения  $stop$  рабочему для начала очистки.

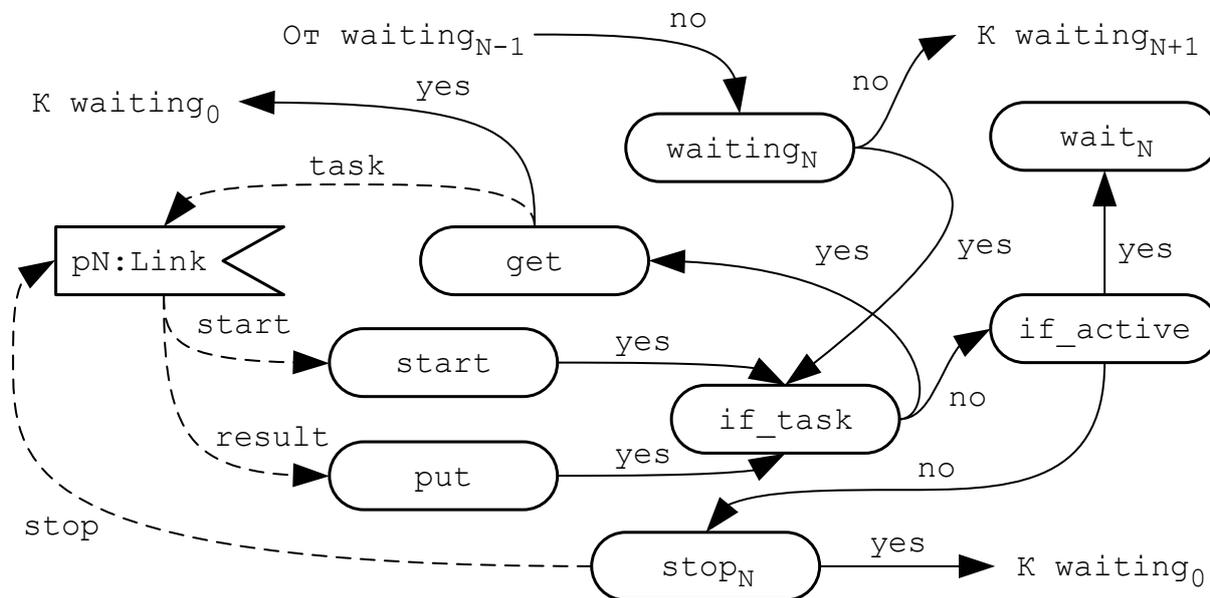


Рисунок 3 — Фрагмент диаграммы управляющего процесса для алгоритма вычисления с портфелем задач

Остановка вычислений и очистка начинаются, если нет задач и нет активных рабочих процессов. Когда сформирована и отправлена очередная задача, происходит поиск ожидающего рабочего процесса. Если такой процесс найдется и есть еще одна задача для обработки, то выполняется ее формирование и отправка рабочему. В противном случае управляющий процесс либо заканчивает обработку сообщения и ждет ответа от рабочих, либо начинает остановку вычислений и очистку.

Также в первой главе рассмотрено построение модели типовых решений «метод переменных направлений» и «конвейер», что демонстрирует адекватность рассмотренного подхода для моделирования вычислительных процессов, возникающих в задачах численного моделирования.

**Во второй главе** приводится обзор алгоритмов обучения гиперрадиальной (HRBF) нейронной сети. Для данной сети строится последовательный алгоритм обучения, затем по последовательному алгоритму с использованием типового решения «портфель задач» синтезируется параллельный алгоритм. Приводятся результаты численного моделирования.

Принципы функционирования гиперрадиальной нейронной сети описываются выражениями (9–13). Изначально на выходные узлы сети подается входной вектор данных  $X_l = (x_{l1}, x_{l2}, \dots, x_{lN})$  для обучения, который является  $l$ -й строкой матрицы обучающих данных  $X$  размера  $L \times N$ , где  $L$  —

количество обучающих пар,  $N$  — количество входных параметров обучения. Данные от этих узлов сдвигаются следующей функцией:

$$v_j^{(k)}(X_l) = x_{lj} - c_j^{(k)}, \quad (9)$$

где  $C^{(k)} = (c_1^{(k)}, c_2^{(k)}, \dots, c_N^{(k)})$  — вектор центров радиальной функции  $k$ -ого нейрона. Далее результат сдвига масштабируется следующей функцией:

$$z_i^{(k)}(X_l) = \sum_{j=1}^N q_{ij}^{(k)} \cdot v_j^{(k)}(X_l), \quad (10)$$

где  $Q^{(k)} = (q_{ij}^{(k)})$  — матрица диаметров радиальной функции  $k$ -ого нейрона размера  $N \times N$ . Затем квадраты масштабированных сигналов суммируются,

$$u^{(k)}(X_l) = \sum_{i=1}^N \left( z_i^{(k)}(X_l) \right)^2, \quad (11)$$

и подаются как аргумент функции активации  $k$ -ого нейрона:

$$f^{(k)}(X_l) = \exp\left(-\frac{u^{(k)}(X_l)}{2}\right). \quad (12)$$

Получаем выход нейронной сети:

$$y(X_l) = w^{(0)} + \sum_{i=1}^K w^{(i)} f^{(i)}(X_l), \quad (13)$$

где  $K$  — количество нейронов сети,  $W = (w^{(0)}, w^{(1)}, \dots, w^{(K)})$  — вектор весов выходного слоя.

Целевая функция, которая подлежит минимизации во время обучения, имеет следующий вид:

$$E(X, D) = \frac{1}{2} \sum_{l=1}^L (y(X_l) - d_l)^2, \quad (14)$$

где  $d_l$  —  $l$ -й элемент вектора отклика  $D$  длины  $L$ ,  $X_l$  и  $d_l$  — это обучающая пара.

Потоковый граф описанной гиперрадиальной нейронной сети (9–13) представлен на рис. 4.

Разработанный алгоритм обучения нейронной сети основан на модифицированном гибридном алгоритме, состоит в использовании метода грубой силы и выполняет обучение и подбор количества нейронов скрытого слоя. Он заключается в итеративном добавлении различающихся случайным положением центра нейронов в идентичные копии нейронных сетей и дообучение этих копий. На каждой итерации алгоритма производится выбор лучшей копии сети (по критерию достижения минимального значения целевой функцией) и её копирование для замены худших копий. Далее итерация добавления повторяется. Критерием остановки алгоритма является

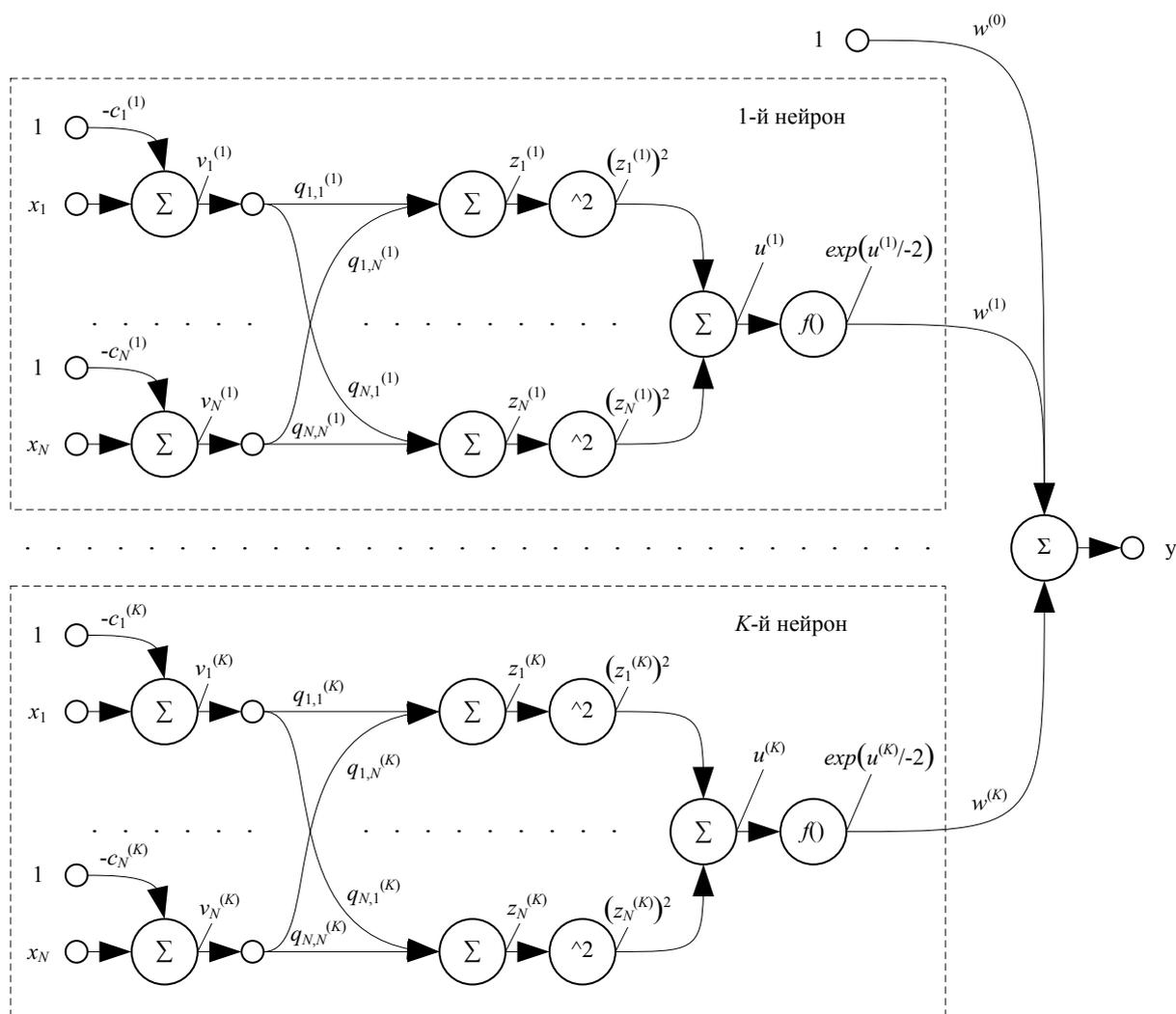


Рисунок 4 — Поточный граф гиперрадиальной нейронной сети

либо достижение заданного предельного количества нейронов, либо достижение заданного значения целевой функции.

Параллельный алгоритм обучения реализован на языке C++ с использованием библиотеки LAPACK. Он заключается в параллельном выполнении процессов добавления нейронов в идентичные копии нейронных сетей и их дообучение. Применение типового решения «портфель задач» рис. 1–3 для распараллеливания состоит в определении методов *get*, *put*, *if\_task*. В «портфеле задач» для управления вычислениями хранятся:

- а)  $K$  — текущее количество нейронов в скрытом слое;
- б)  $K_{max}$  — максимальное число нейронов в скрытом слое;
- в)  $W_1$  — число выданных задач,  $W_2$  — решенных задач;
- г)  $W_{max}$  — максимальное количество задач (попыток добавления нейрона);
- д)  $E_{end}$  — значение, при достижении целевой функцией которого процесс обучения останавливается;

- е)  $\Psi_K$  — коэффициенты обученной нейронной сети с  $K$  нейронами (включают вектора центров, матрицы диаметров и вектор весов выходного слоя);
- ж)  $\Psi_{K+1}$  — коэффициенты обученной нейронной сети с  $K + 1$  нейронами;
- з)  $E_{K+1}$  — значение целевой функции сети  $\Psi_{K+1}$ .

Задача определяется коэффициентами  $\Psi'_K$  и положением центра добавляемого нейрона  $\psi'$ . Результат ее вычисления — коэффициенты  $\Psi'_{K+1}$  и значение ошибки  $E'_{K+1}$ . Как и в последовательном случае, при решении задачи производится уточнение коэффициентов сети  $\Psi'_{K+1}$  гибридным алгоритмом.

В начальном состоянии (*start*) определяются постоянные алгоритма  $K_{max}$ ,  $W_{max}$ ,  $E_{end}$ ; переменные  $W_1 := 0$ ,  $W_2 := 0$ ,  $K := 2$ ,  $\Psi_K := \Psi_2$  (параметры сети с двумя нейронами),  $\Psi_{K+1}$  — не определены,  $E_{K+1} \gg E_{end}$ .

*Проверка наличия задачи (if\_task)*. Возможна выборка очередной задачи на обработку, если одновременно выполняются следующие условия. Не достигнута заданная ошибка обучения:  $E_{K+1} > E_{end}$ ; Остались не выданные задачи:  $W_1 < W_{max}$ .

*Алгоритм извлечения задачи (get)*. Присвоить  $\Psi'_K := \Psi_K$ ; определить случайное положение центра нейрона  $\psi'$ . Увеличить счетчик запущенных задач  $W_1 := W_1 + 1$ .

*Алгоритм помещения результата вычисления задачи (put)*. Если  $E'_{K+1} < E_{K+1}$ , то определить  $E_{K+1} := E'_{K+1}$  и  $\Psi_{K+1} := \Psi'_{K+1}$ . Присвоить  $W_2 = W_2 + 1$ . Если  $W_2 = W_{max}$ , то закончено обучение слоя. В этом случае, если  $K < K_{max}$ , переходим к следующему слою:  $K := K + 1$ ;  $\Psi_K := \Psi_{K+1}$ ,  $W_1 := 0$ ;  $W_2 := 0$ .

Алгоритм завершается, когда будут обучены все сети с числом нейронов до  $K_{max}$  включительно, либо будет достигнуто заданное значение ошибки обучения  $E_{K+1} \leq E_{end}$ .

Эксперимент по обучению нейронной сети параллельным алгоритмом на основе типового решения «портфель задач» был проведен на кластере Сергей Королёв (<http://hpc.ssau.ru/>). Запуск задач производился на узлах 3-х типов с 3-мя различными типами процессоров соответственно. Ниже приведены результаты эксперимента на одном из типов узлов с процессорами типа Intel®Xeon®E5-2665 (20M Cache, 2.40 GHz, 8.00 GT/s Intel®QPI), имеющими 16 вычислительных ядер.

В рамках данного эксперимента на быстродействие влияло количество доступных вычислительных ядер на узле, остальные параметры ощутимого влияния не оказывали.

В проведённых вычислительных экспериментах обучение сети выполнялось на выборках котировок фондовых рынков. Условия экспериментов состояли в следующем.

- а) Варьировалось максимальное количество рабочих процессов  $P$  «портфеля задач» (объявлено в конкретном эксперименте).

- б) Предельное количество нейронов  $K_{max}$  во всех экспериментах равнялось 100.
- в) Количество попыток добавления нейронов  $W_{max}$  во всех экспериментах равнялось 100.

Для осуществления сравнительного анализа были произведены замеры времени исполнения последовательного и параллельного алгоритмов. Для минимизации воздействия сторонней динамической нагрузки процессоров на измеряемое время, производилось по 20 запусков. При этом, полученное время исполнения параллельного алгоритма  $t_{pr_i}(P)$  было усреднено:

$$t_{pr}(P) = \frac{\sum_{i=1}^{20} t_{pr_i}(P)}{20}. \quad (15)$$

Для времени исполнения последовательного алгоритма  $t_{sq_i}$ , было найдено минимальное значение  $t_{sq_{min}} = \min_{i=\overline{1,20}} t_{sq_i}$ . И на его основе рассчитано эталонное время исполнения параллельного алгоритма

$$t_{std}(P) = \frac{t_{sq_{min}}}{P}. \quad (16)$$

Средне-квадратичное отклонение рассчитывалось для  $t_{pr_i}(P)$ .

На узлах с процессорами Intel®Xeon®E5-2665 максимальное количество рабочих процессов изменялось в интервале  $P = \overline{1,40}$ . Данные, полученные при проведении эксперимента, приведены на рис. 5, соответствующее средне-квадратичное отклонение изображено на рис. 6.

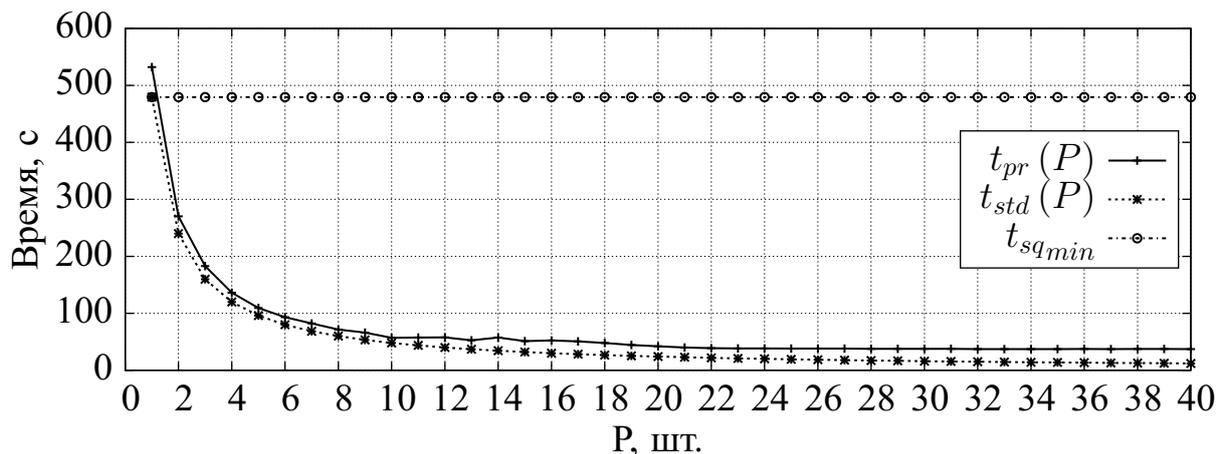


Рисунок 5 — Зависимость времени исполнения от количества рабочих процессов (16-ядер)

Отметим, что данный алгоритм является распределенным по построению и может исполняться как в системах с разделяемой, так и в системах с распределенной памятью. Техника его распараллеливания является универ-

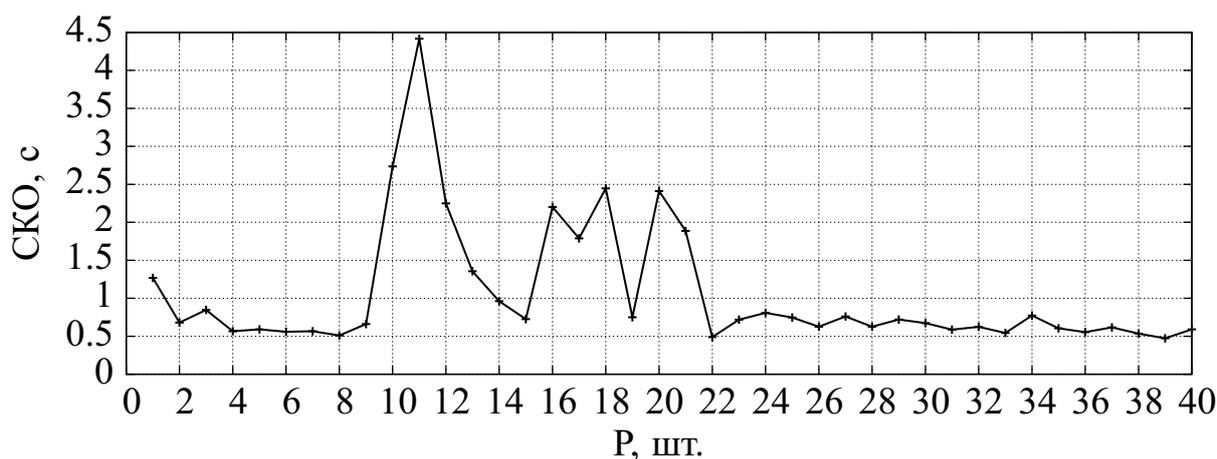


Рисунок 6 — Зависимость СКО от количества рабочих процессов (16-ядер)

сальной и может применяться для решения подобных задач с использованием типового решения «портфель задач».

**В третьей главе** описываются программный комплекс численного моделирования на основе типовых решений и результаты его применения для ряда задач математического моделирования.

Для типового решения «портфель задач» описаны разработанные среды многопоточного исполнения на основе Windows API, POSIX и MPI, позволяющие проводить высокопроизводительные вычисления на системах с общей и распределённой памятью.

Оценено снижение трудоёмкости при построении параллельных численных методов на основе типовых решений путём измерения объемов ручного и автоматического кодирования.

Показано, что программный комплекс, наряду с преимуществами в скорости разработки и возможностью автоматически запускать исходную последовательную программу на разных высокопроизводительных вычислительных системах, обеспечивает достаточно эффективное распараллеливание кода.

Реализованные в диссертационной работе типовые решения входят в состав программного комплекса Templet Web, выполненного в виде web-сервиса для автоматизации параллельных вычислений и размещенного по адресу <http://templet.ssau.ru>.

**В заключении** сформулированы основные выводы и результаты, полученные в работе:

- а) Выполнен анализ актуальных проблем разработки программных комплексов высокопроизводительных научных вычислений. Показана перспективность развития подхода, основанного на применении типовых решений.
- б) Разработана математическая модель вычислительного процесса для

описания различных типовых решений в области высокопроизводительных научных вычислений.

- в) Разработана графическая модель алгоритма для типового решения «портфель задач» на основе математической модели вычислительного процесса и графической нотации Templet.
- г) Предложена методика синтеза параллельного алгоритма на примере алгоритма обучения гиперрадиальной нейронной сети с использованием модели типового решения «портфель задач».
- д) Разработан и внедрён в учебно-исследовательский процесс программный комплекс «портфель задач» для использования на высокопроизводительных вычислительных системах в составе сервиса Templet Web.
- е) Проведены экспериментальные исследования, подтверждающие эффективность применения программного комплекса. В частности, в задаче обучения гиперрадиальной нейронной сети ускорение составило порядка 13 раз на 16-ти ядерном узле при этом снижение сложности разработки, оцененное по объёму ручного кодирования, составило 2,34 раза.

**Основные результаты диссертации опубликованы в следующих работах:**

***Статьи в изданиях, рекомендованных ВАК:***

- 1) **Литвинов, В.Г.** Разработка и применение вычислительной модели типовых решений: пример использования «портфеля задач» для обучения нейронной сети HRBF [Текст] / **В.Г. Литвинов** // Вестник Самарского гос. техн. ун-та. Серия: Физ.-мат. науки. 2014. №4(25). С. 146–153.
- 2) Востокин, С.В. Применение комплекса параллельного программирования Graphplus templet в моделировании [Текст] / С.В. Востокин, **В.Г. Литвинов**, А.Р. Хайрутдинов // Программные продукты и системы. 2012. №3(99). С.12–16.
- 3) Востокин, С.В. Программный комплекс параллельного программирования Graphplus templet [Текст] / С.В. Востокин, **В.Г. Литвинов**, А.Р. Хайрутдинов // Вестник Самарского гос. техн. ун-та. Серия: Физ.-мат. науки. 2011. №4(25). С. 146–153.

***Статьи в других изданиях:***

- 4) Востокин, С.В. Применение типовых решений для распараллеливания алгоритмов численного моделирования [Текст] / С.В. Востокин, **В.Г. Литвинов** // Перспективные информационные технологии (ПИТ 2013): Тр. Международной научно-технической конференции / под ред. С.А. Прохорова. Самара. 2013. С.184–186.
- 5) Востокин, С.В. Представление алгоритмов высокопроизводительных вычислений в нотации Templet [Текст] / С.В. Востокин, **В.Г. Литвинов**, Д.Д. Макагонова, А.Р. Хайрутдинов // Перспективные информационные технологии в научных исследованиях, проектировании и обучении (ПИТ-2012): Тр. на-

учно-технической конференции с международным участием. Самара. 2012. С.74–76.

6) Востокин, С.В. Визуальное моделирование параллельных алгоритмов в процессно-ориентированной нотации Templet [Текст] / С.В. Востокин, **В.Г. Литвинов**, Д.Д. Макагонова, А.Р. Хайрутдинов // Параллельные вычисления и задачи управления: Тр. Шестой Международн. конф-ции. М.: ИПУ РАН, 2012. Том 1. С.260–269.

7) **Литвинов, В.Г.** Моделирование динамики пространственно-распределенных систем типа «реакция-диффузия» с использованием паттерна «портфель задач» на суперкомпьютере [Текст] / **В.Г. Литвинов** // Всероссийская молодежная научная конференция с международным участием «XI Королёвские чтения». Самара, СГАУ. 2011. С.293.

8) **Литвинов, В.Г.** Разработка комплекса типовых решений параллельных вычислений на основе модели «Граф Плюс» на суперкомпьютерных системах [Текст] / **В.Г. Литвинов** // Тезисы докладов XXXVII Самарской областной студенческой научной конференции. Часть I. Самара. 2011. С.275.

9) **Литвинов, В.Г.** Методы построения оптимальной структуры HRBF нейронной сети для решения задачи прогнозирования [Текст] / **В.Г. Литвинов**, О.П. Солдатова // Перспективные информационные технологии для авиации и космоса (ПИТ-2010). Труды Международной конференции с элементами научной школы для молодежи. Самара. 2010. С.252–254.

10) **Литвинов, В.Г.** Экстраполяция поведения фондового рынка искусственной нейронной сетью на основе радиально базисных функций [Текст] / **В.Г. Литвинов** // Тезисы докладов XXXVI самарской областной студенческой научной конференции, Самара. 2010. С.307–308.

11) **Литвинов, В.Г.** Нейросетевой аппроксиматор на основе радиально-базисных функций с масштабирующей матрицей [Текст] / **В.Г. Литвинов** // Всероссийская молодежная научная конференция с международным участием «X королевские чтения». Самара. 2009. С.293.

***Свидетельства о регистрации программных комплексов:***

12) **Литвинов, В.Г.** Программа обучения и оптимизации структуры гиперрадиальной нейронной сети с использованием типового решения «портфель задач». // Свидетельство о государственной регистрации программ для ЭВМ №2014661813, выданное Федеральной службой по интеллектуальной собственности. Зарегистрировано в Реестре программ для ЭВМ 13.11.2014.



Подписано в печать 1.09.2015 г.  
Формат 60 x 84 1/16. Бумага ксероксная. Печать оперативная.  
Объем – 1,25 усл. печ. л. Тираж 100 экз. Заказ № 61.

Отпечатано в типографии ООО «Инсома-пресс»  
443080, г. Самара, ул. Санфировой, 110А, оф. 22А,  
тел. 222-92-40, E-mail: insoma@bk.ru