

СРЕДСТВА АВТОМАТИЗАЦИИ ПАРАЛЛЕЛЬНОГО РЕШЕНИЯ КОМБИНАТОРНЫХ ЗАДАЧ В ГЕТЕРОГЕННОЙ РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

Колосов А.Д.

*ФБГУН «Институт динамики систем и теории управления имени В.М. Матросова СО РАН», Иркутск, Россия
(664033, г. Иркутск, ул. Лермонтова, 134), e-mail: akolosov.irk@gmail.com*

В статье рассматривается сервис-ориентированный подход к автоматизации проведения многовариантных расчетов при параллельном решении комбинаторных задач в гетерогенной распределенной вычислительной среде. В качестве примера рассматривается создание сервиса для решения задачи коммивояжера с помощью рекуррентной нейронной сети и ее параллельной реализацией на графическом процессоре с использованием технологии CUDA Nvidia. Сервис используется для параллельного проведения многовариантных расчетов при подборе параметров нейронной сети и конфигурации запуска нитей графического процессора. Рассматриваются функциональные возможности и аспекты программной реализации сервиса с помощью инструментальных средств, разработанных в Институте динамики систем и теории управления имени В.М. Матросова СО РАН. Применение этих средств для реализации сервиса обеспечивает удаленный запуск задачи, создание пакета заданий для проведения многовариантных расчетов, подбор подходящих вычислительных узлов распределенной среды, постановку заданий в очередь к системе управления заданиями выбранного узла, мониторинг выполнения заданий, постобработку результатов решения, уведомление пользователя о завершении выполнения. Приводятся результаты применения сервиса для решения ряда задач.

Ключевые слова: распределенные вычислительные среды, технология CUDA, параллельные алгоритмы.

THE TOOLKIT FOR AUTOMATION OF PARALLEL SOLVING OF COMBINATORIAL PROBLEMS IN HETEROGENEOUS DISTRIBUTED COMPUTING ENVIRONMENT

Kolosov A.D.

*Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences,
Irkutsk, Russia (664033, Irkutsk, Lermontov str., 134), e-mail: akolosov.irk@gmail.com*

The article discusses the service-oriented approach to the automation of multivariate calculations in parallel combinatorial problems in a heterogeneous distributed computing environment. The creating of the service for solving the traveling salesman problem using recurrent neural network and its implementation on GPU with CUDA Nvidia technology is given in the article as an example. This service used for the parallel multivariate calculations for selecting of parameters of the neural network and the grid configuration of GPU. We consider the features and aspects of software implementation services using tools developed at the Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences. The use of these software to implement the service provides remote task, the creating a package of jobs for multivariate calculations, selection of appropriate computing nodes of the distributed environment, the sending of this package to the resource manager of the selected node, monitoring the jobs, post-processing solutions, user notification of completion. The results of the use of the service for a number of examples are described.

Keywords: distributed computing environments, CUDA technology, parallel algorithms.

В настоящее время представляется актуальным создание инструментальных средств, обеспечивающих разработку программных сред для доступа предметных специалистов к высокопроизводительным вычислительным ресурсам и использования этих ресурсов без необходимости углубленного знания вычислительных архитектур и низкоуровневых средств разработки приложений. Одновременно прослеживаются тенденции, с одной стороны, использования сервис-ориентированного подхода, подразумевающего организацию сервисов доступа к вычислительным ресурсам посредством сети Интернет, с другой стороны,

оформления функций приложения пользователя в виде сервиса.

Целью проводимых исследований являлась разработка сервис-ориентированных средств для автоматизации проведения вычислительных экспериментов при параллельном решении комбинаторных задач в гетерогенной распределенной вычислительной среде с применением технологии CUDA Nvidia и рекуррентных нейронных сетей, активно используемых для их решения в настоящее время. Подбор таких параметров функции энергии сети, при которых достигается хорошая устойчивость сети и близость к оптимальному значению результата, а также параметров сетки grid графического процессора (graphics processing unit, GPU) для повышения эффективности времени выполнения приводит к необходимости проведения многовариантных расчетов. Такие вычислительные эксперименты являются не только ресурсоемкими, но и требуют больших трудозатрат пользователя. Эти обстоятельства актуализируют создание инструментария, сокращающего время выполнения расчетов и повышающего эффективность работы пользователя.

В докладе рассматриваются функциональные возможности сервиса, автоматизирующего проведение вычислительных экспериментов при параллельном решении комбинаторных задач на GPU с помощью рекуррентных нейронных сетей и с использованием технологии CUDA, описываются аспекты программной реализации сервиса путем применения инструментальных средств HrcSoMaS [1]. Приводятся результаты тестирования этого сервиса на примере решения задачи о коммивояжере с помощью искусственной нейронной сети (ИНС) Вана [9] в гетерогенной вычислительной среде.

Рекуррентная нейронная сеть

Отдельную группу ИНС составляют сети с обратными связями между различными слоями нейронов. Это так называемые рекуррентные сети. Их общая черта — передача сигналов с выходного или скрытого слоя во входной слой. Главная особенность таких сетей — динамические зависимости на каждом этапе функционирования. У сетей без обратных связей нет памяти, их выход полностью определяется текущими входами и значениями весов. В отличие от сетей прямого распространения, которые выдают результат за определенное количество шагов, результатом рекуррентной сети является устойчивое состояние. Примером рекуррентной сети может служить ИНС Вана.

Структура сети Вана подробно описывается в работе [9]. Сходимость и устойчивость сети рассмотрена в работах [5; 10]. Сеть Вана широко применяется для решения различных комбинаторных задач, задач оптимизации. С помощью данной нейронной сети решаются также некоторые задачи, возникающие при организации распределенных вычислений. Например, с использованием нейронной сети Вана успешно решаются задачи о назначении [5; 9]. Задача маршрутизации может быть сведена к задаче размещения [6].

Нейронная сеть Вана описывается дифференциальным уравнением [9]:

$$\frac{du_{ij}^t}{dt} = -\eta \sum_{l=1}^n x_{il}^t - \eta \sum_{k=1}^n x_{kj}^t + 2\eta - \lambda c_{ij} e^{-\frac{t}{\tau}}, \quad (1)$$

где
$$x_{ij} = \frac{1}{1+e^{-\beta u_{ij}}}. \quad (2)$$

Данное уравнение решается численным методом. Одним из самых простых в реализации является метод Эйлера. Для решения уравнения методом Эйлера строится следующий итерационный процесс:

$$u_{ij}^{t+1} = u_{ij}^t - \Delta t \left[\eta (\sum_{l=1}^n x_{il}^t + \sum_{k=1}^n x_{kj}^t - 2) + \lambda c_{ij} e^{-\frac{t}{\tau}} \right], \quad (3)$$

где Δt – шаг по времени; t – момент времени; c_{ij} – стоимость назначения элемента i в позицию j ; Δt , η , λ , τ , β – коэффициенты, подбираемые экспериментально. Подбор коэффициентов существенно влияет на скорость сходимости сети и качество решения.

Решение уравнения сети Вана можно ускорить, применив принцип «Победитель получает всё» (Winner takes all, WTA). Согласно этому принципу, активным остается нейрон с самым большим выходом, остальные нейроны в строке и столбце становятся неактивными [6]. В данной работе нейронная сеть Вана применяется для решения задачи коммивояжера. Задача коммивояжера – классическая задача комбинаторной оптимизации. Целью является нахождение гамильтонова цикла с минимальной суммарной стоимостью во взвешенном полностью связном графе. Эта задача выбрана для тестирования сервиса, автоматизирующего проведение вычислительных экспериментов с ИНС на GPU, т.к. для нее существует большой набор тестов с известным оптимальным решением.

Для ускорения ИНС Вана была распараллелена на GPU с применением технологии CUDA. Нейросетевая динамика рекуррентной нейронной сети Вана состоит в многократном циклическом пересчете матрицы весовых коэффициентов и матрицы состояния сети. К каждому элементу матрицы применяется одинаковый набор инструкций, что соответствует архитектуре SIMD (Single Instruction, Multiple Data). GPU хорошо подходит для реализации параллелизма по данным. Каждому потоку достаётся по одному элементу матрицы для обработки. Это позволяет получить заметное ускорение по сравнению с последовательной версией. Перенос данных CPU-GPU минимален, так как матрицы переносятся на GPU однократно при запуске программы, больше в алгоритме нет затратных операций обмена. Алгоритм WTA в программе выполняется последовательно в силу своей «последовательной природы». Кроме того, выполнение WTA занимает менее одного процента от времени работы программы. Программная реализация алгоритма на языке C++ с использованием технологии CUDA названа в статье gtsp.

Разработка сервиса

Сервис обеспечивает следующие возможности, поддерживаемые инструментарием HpcSoMas Framework:

- Web-интерфейс для запроса пользователя на решение задачи. Задание параметров для проведения вычислительного эксперимента: входные и выходные данные программы, реализующей нейросетевой алгоритм, наблюдаемые выходные параметры, на основании значения которых делается вывод об эффективности решения для заданных входных параметров.
- Прозрачное для пользователя параллельное решение для различных вариантов входных параметров в распределенной вычислительной среде.
- Информирование пользователя об окончании решения. Обработка и визуализация результатов решения.

Сервис для решения задачи коммивояжера (TSP) содержит следующие группы входных параметров (рис. 1):

- Параметры Δt , η , λ , τ , β , из формул (1)-(3), которые могут варьироваться в процессе вычислительного эксперимента. Эти параметры могут быть заданы одним значением (заполняется только крайнее левое поле ввода на форме, представленной на рис. 1), в этом случае сервис подбирает подходящий ресурс для запуска задания пользователя и последующих этапов процесса выполнения. Эти параметры могут так же быть заданы интервалом изменения значений от нижней границы до верхней с определенным шагом. При этом сервисом организуются многовариантные вычисления с целью подбора такой комбинации параметров, при которой будет достигнуто оптимальное решение. В этом случае сервис формирует набор заданий, подбирает подходящие вычислительные ресурсы, запускает задания (в зависимости от загруженности ресурса) или ставит в очередь к соответствующей системе управления заданиями.
- Параметры gridXY и blockXY, определяющие структуру сетки (grid) для GPU.
- Параметры, задающие входные данные – размерность задачи (количество городов), расстояния между городами, количество итераций, предотвращающее заикливание алгоритма при попытке улучшить решение.
- Параметры, задающие постобработку выходных данных, называемых наблюдаемыми параметрами. Эти параметры используются для получения статистических данных: средней длины маршрута, среднего и общего времени выполнения всего задания, среднего и общего времени выполнения отдельных этапов и т.д. Вид обработки включает нахождение среднего, минимального, максимального или суммарного значения для наблюдаемого параметра.

- Параметр сервиса – количество повторных запусков для получения статистики по наблюдаемым параметрам.

сервисы задания решения ресурсы

Укажите имя задания

Коэффициенты ИНС :

0.01	0	0	параметр Δt
1	0	0	параметр η
1	0	0	параметр λ
1	1000	1	параметр τ
0.1	0	0	параметр β

* от, до и шаг, можно указывать только начальное значение

1000 число итераций
10 повтор вычислений для приближения к опт. значению

матрица весов:
Выберите файл | Файл не выбран | загрузить
статус

10 размерность задачи

Настройки GPU :
1 gridXY 10 blockXY

Настройки постобработки :
cost наблюдаемый параметр
average выбор вида обработки результатов
1 количество запусков

Создать задание

Лаборатория параллельных и распределенных вычислительных систем,
ИДСТУ СО РАН, 2015

Рис. 1. Web-интерфейс сервиса для решения задачи TSP с помощью ИНС на GPU

На этапе обработки данных для решения задачи коммивояжера можно визуализировать путь с помощью сервиса обработки. Ниже приведен пример визуализации для задачи небольшой размерности, 10 городов. Пример взят из работы [6]. С помощью варьирования параметра Δt постепенно получает оптимальный маршрут. На рис. 2 показаны субоптимальный и оптимальный маршруты.

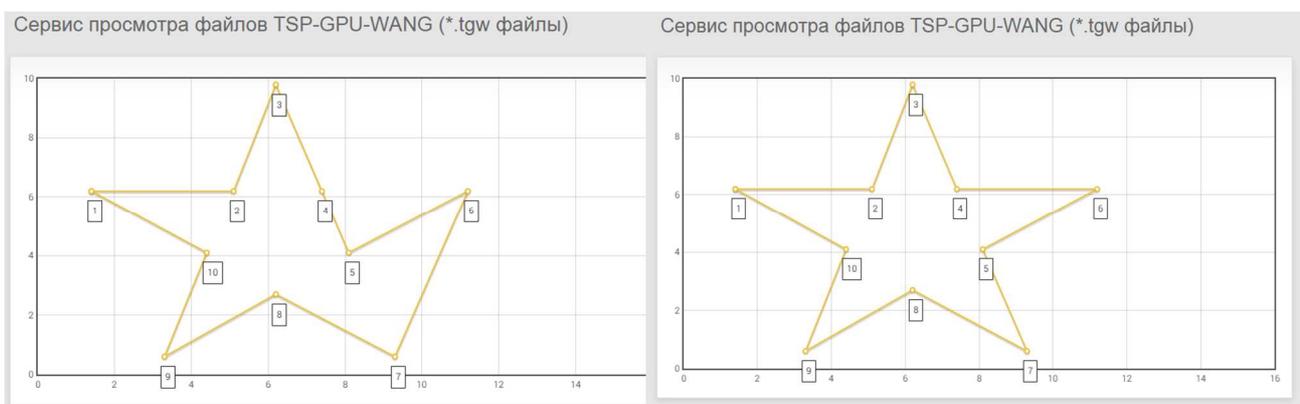


Рис. 2. Визуализация маршрутов с помощью сервиса

Инструментарий HpcSoMas Framework

Для реализации сервиса использованы инструментальные средства (ИС) HpcSoMaS Framework [1], предназначенные для разработки мультиагентных систем, автоматизирующих процесс управления распределенными вычислениями. Особенностью таких систем является реализация агентов в виде сервисов. ИС включают средства распределения ресурсов, обеспечивающие выбор наименее загруженного вычислительного кластера (ВК), отвечающего требованиям пользователя, и средства организации взаимодействия с системой управления заданиями (СУПЗ) ВК. В состав ИС HpcSoMaS Framework входят: средства создания агентов на базе нейронных сетей; библиотека разработки сервисов на основе стандарта REST (Representational State Transfer) и протокола SOAP (Simple Object Access Protocol), а также готовые сервисы, реализующие базовые функции агентов, созданные на основе библиотечных классов и требующие для своего использования только конфигурационную настройку; графические средства проектирования сервисов; документация по используемым форматам файлов конфигурации сервисов.

Основными компонентами в HpcSoMaS Framework являются агенты-менеджеры, которые и производят основную работу по распределению заданий с учетом их классификации [3], подбор наилучшей вычислительной среды под конкретную задачу, анализ состояний СУПЗ, перераспределение заданий, их декомпозицию. Распределение вычислительных ресурсов осуществляется на основе экономического механизма регулирования их спроса и предложения [2]. Данные агенты реализованы как сервлеты с помощью средств Java EE. Для выполнения различных функций на вычислительных кластерах, зависящих от конфигурации ВК, они запускают агентов, реализованных как сервисы на основе протокола soap.

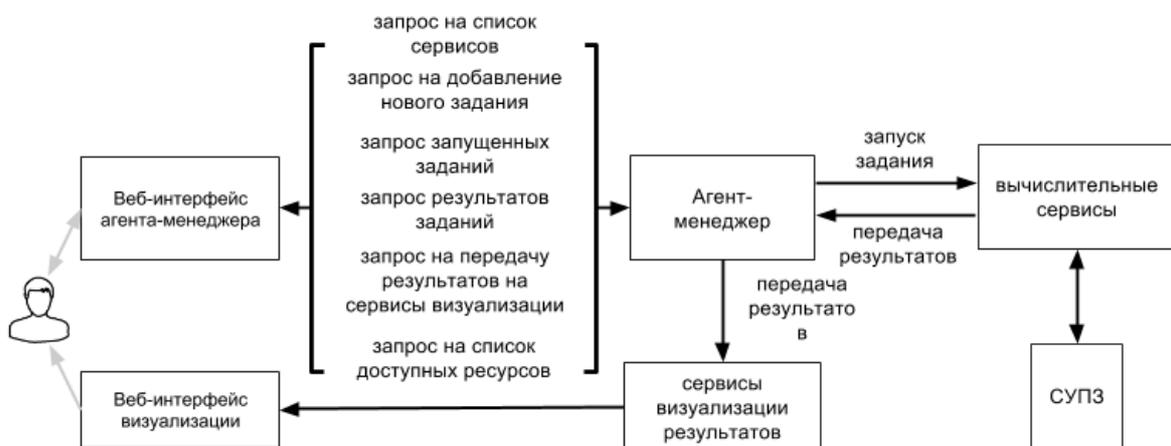


Рис. 3. Схема решения задачи с использованием сервиса

Этапы проведения вычислительного эксперимента с помощью рассматриваемого сервиса приведены на рис. 3. Агент-менеджер через свой web-интерфейс получает и

обрабатывает запросы, запуская для обработки вычислительные сервисы и сервисы визуализации результатов.

Вычислительные эксперименты

При создании сервиса в качестве распределенной вычислительной среды была использована разнородная кластерная Grid-система ИДСТУ СО РАН, включающая: однородный ВК Blackford с SMP-узлами, неоднородный ВК «Академик В.М. Матросов» с SMP-узлами и узлом с графическими процессорами NVidia C2070 (Fermi); неоднородный ВК Tesla, в состав которого входит 4 четырехъядерных процессора и 8 GPU NVidia Tesla C1060. Для тестирования сервиса были выбраны задачи из библиотеки TSPLIB [7], для которых в работах [4; 8] были приведены результаты, полученные с помощью других алгоритмов.

Эксперимент проводился следующим образом. С помощью сервиса варьировались коэффициенты, используемые в формулах (1)-(3), затем с этими коэффициентами производилась серия запусков решения каждой задачи с нахождением средней длины пути для оценки качества полученного решения. Сервис показал работоспособность и экономию трудозатрат при проведении экспериментов.

Результаты экспериментов запусков gtsp сравнивались с результатами, приведенными в [4; 8] для известного «муравьиного» алгоритма AS (Ant System), обозначенного как AS-CPU, и его параллельной реализации на GPU (AS-GPU). На рис. 4, 5 приведены результаты сравнения достигнутого сервисом качества решения ряда задач с помощью разработанного решателя gtsp с соответствующими данными, приведенными в [4; 8]. Качество (аналогично [10]) определялось как отношение оптимального значения к полученному в результате работы тестируемого алгоритма значению. С помощью функций постобработки сервиса выбиралось среднее значение для заданного числа запусков.

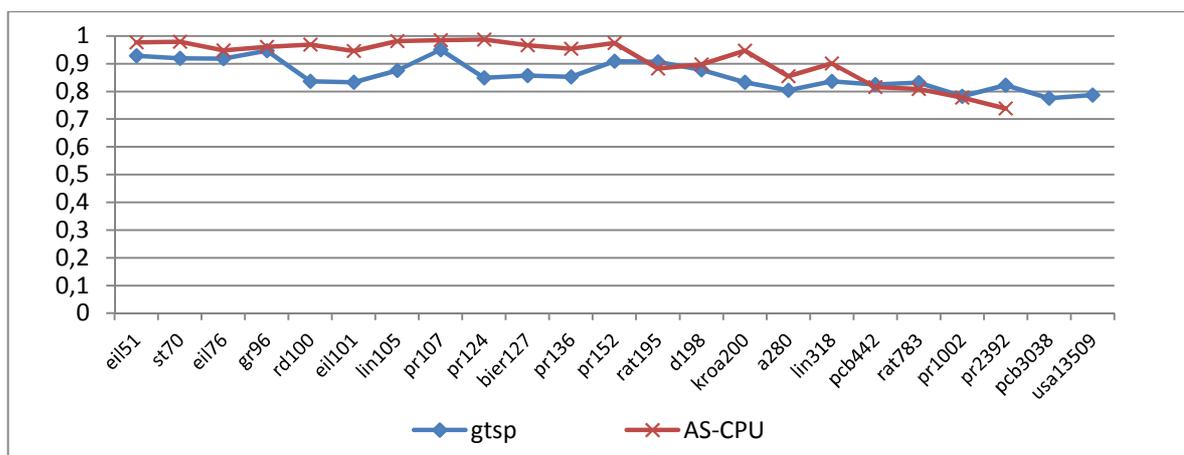


Рис. 4. Сравнение качества решения gtsp и AS-CPU

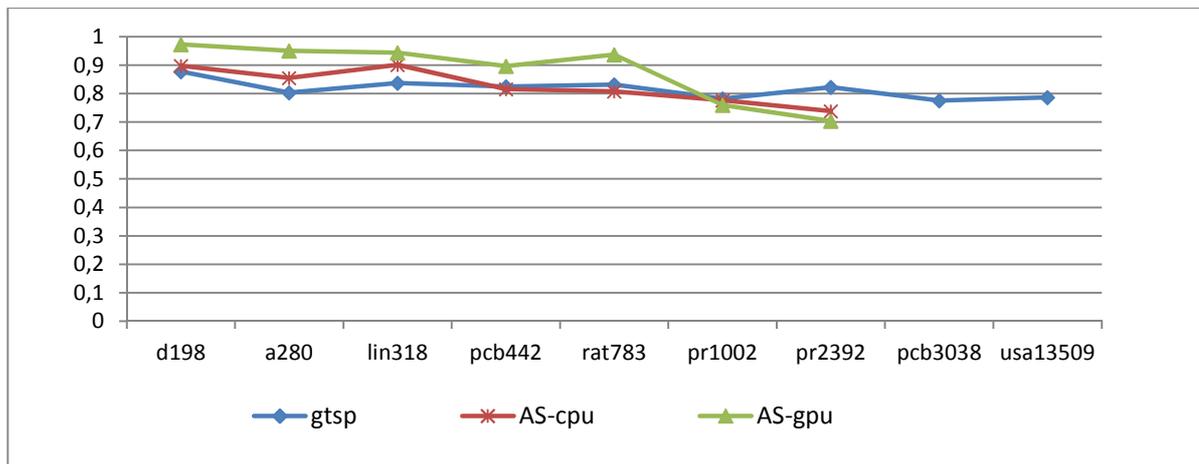


Рис. 5. Сравнение результатов качества работы трех алгоритмов

В табл. 1 приведены усредненные данные для сравнения времени выполнения последовательной и параллельной версий для 100 запусков. Из табл. 1 видно, что ускорение начинает возрастать для реализаций на GPU с увеличением размерности задачи. При большем количестве итераций качество решения, представленное на рис. 4 и рис. 5, получаемое с помощью gtsp, может быть улучшено, иногда существенно.

Таблица 1

Среднее время выполнения и среднее ускорение для 100 запусков

Примеры	Размерность	Общее время решения, в мсек.			Ускорение	
		AS-CPU	AS-GPU	gtsp	AS-GPU	gtsp
d198	198	2080,72	263,91	255,96	7,89x	8,13x
a280	280	4844,59	505,51	529,95	9,58x	9,14x
lin318	318	9797,03	897,29	979,79	10,92x	10x
pcb442	442	18534,37	1153,95	1467,74	16,06x	12,63x
rat783	783	123220,58	5673,15	5315,48	21,72x	23,18x
pr1002	1002	381949,72	8706,18	11633,98	43,87x	32,83x
pr2392	2393	5867605,87	208478,2	163687,9	28,14x	35,85x

Заключение

В целом можно сделать вывод, что рассматриваемые в статье программные средства показали свою работоспособность и эффективность для задач достаточно большой размерности. Дальнейшее направление исследований связано с улучшением качества решения, оптимизацией использования GPU и применением представленного подхода для решения новых задач.

Исследование выполнено при финансовой поддержке РФФИ, проект № 15-29-07955-офи_м.

Список литературы

1. Бычков И.В., Опарин Г.А., Феоктистов А.Г., Богданова В.Г., Пашинин А.А. Мультиагентные методы и инструментальные средства управления в сервис-

ориентированной распределенной вычислительной среде // Труды Института системного программирования РАН. – 2014. – Т. 26. – Вып. 5. – С. 65-82.

2. Бычков И.В., Опарин Г.А., Феоктистов А.Г., Кантер А.Н. Мультиагентный алгоритм распределения вычислительных ресурсов на основе экономического механизма регулирования их спроса и предложения // Вестник компьютерных и информационных технологий. – 2014. – № 1. – С. 39-45.

3. Бычков И.В., Опарин Г.А., Феоктистов А.Г., Корсуков А.С. Распределение заданий в интегрированной кластерной системе на основе их классификации // Вычислительные технологии. – 2013. – Т. 18. – № 2. – С. 25–32.

4. Dawson L., and Stewart I.A. Improving Ant Colony Optimization performance on the GPU using CUDA // IEEE Congress on Evolutionary Computation. – 2013. – P. 1901-1908.

5. Hung D.L., Wang J. Digital hardware realization of a recurrent neural network for solving the assignment problem // Neurocomputing. – 2003. – Vol. 51. – P. 447-461.

6. Siqueira P.H., Steiner M.T.A., Scheer S. A new approach to solve the travelling salesman problem // Neurocomputing. – 2007. – Vol. 70. – P. 1013-1021.

7. TSPLIB Test Data. – Режим доступа: <http://www.math.uwaterloo.ca/tsp/data> (дата обращения: 12.09.2015).

8. Uchida A., Ito Y., Koji N. An Efficient GPU Implementation of Ant Colony Optimization for the Traveling Salesman Problem // International Conference on Computing, Networking and Communications (ICNC). – 2012. – P. 94-102.

9. Wang J. An analog neural network for solving the assignment problem, Electronic Letters. – 1992. – Vol. – 28. – № 11. – P. 1047–1050.

10. Wang J. Analysis and design of a recurrent neural network for linear programming // IEEE Trans. Circuits Syst. I: Fundamental Theory Applicat. – 1993. – Vol. 40. – P. 613–618.

Рецензенты:

Опарин Г.А., д.т.н., профессор, зам. директора по научной работе ФБГУН «Институт динамики систем и теории управления» СО РАН, г. Иркутск;

Ружников Г.М., д.т.н., с.н.с., зав. отделением «Информационные технологии и системы» ФБГУН «Институт динамики систем и теории управления» СО РАН, г. Иркутск.