УДК 681.3.06

МАСШТАБИРУЕМОЕ ПРИЛОЖЕНИЕ ДЛЯ ПОИСКА ГЛОБАЛЬНЫХ МИНИМУМОВ МНОГОЭКСТРЕМАЛЬНЫХ ФУНКЦИЙ*

И. В. Бычков[1], Г. А. Опарин[1], А. Н. Черных[2], А. Г. Феоктистов[1], С. А. Горский[1], Р. Ривера-Родригес[2]

[1] *Институт динамики систем и теории управления им. В. М. Матросова СО РАН, 664033, Россия, г. Иркутск, ул. Лермонтова, 134, а/я 29*

[2] *Centro de investigación Cientifica y de educatión Superior de Ensenada, Baja California, 22860, Mexico, Ensenada, Ensenada-Tijuana Highway, 3918, postbox 360*

E-mail: chernykh@cicese.mx

agf@icc.ru

Рассмотрена актуальная проблема обеспечения масштабируемости вычислений при решении многоэкстремальных задач, возникающих в различных областях научных исследований, включая обработку изображений. Предложен подход к разработке масштабируемого приложения Градиент для решения задачи глобальной оптимизации многоэкстремальных функций с помощью метода мультистарта в инструментальном комплексе Orlando. Реализован дополнительный этап вычислений в схеме решения задачи, позволяющий осуществить декомпозицию задачи с учётом производительности вычислительных ресурсов и тем самым обеспечить минимизацию времени её решения по сравнению с классическим методом мультистарта. Разработаны специальные агенты системы метамониторинга для измерения производительности ресурсов относительно решаемой задачи.

Ключевые слова: распределённые вычисления, масштабируемое приложение, многоэкстремальные функции.

DOI: 10.15372/AUT20180113

Введение. Стремительное развитие высокопроизводительных систем ведёт к количественному увеличению их вычислительных элементов (узлов, процессоров и ядер), в связи с чем обеспечение масштабируемости вычислений является актуальным [1]. Решение этой нетривиальной проблемы во многом зависит от стратегии декомпозиции задачи на подзадачи в зависимости от числа доступных ресурсов [2]. В суперкомпьютерных центрах коллективного пользования выбор данной стратегии обусловлен возможностями систем управления ресурсами, установленными в вычислительных узлах гетерогенной распределённой вычислительной среды (ГРВС).

Типичная задача, требующая привлечения средств высокопроизводительных вычислений, — поиск глобальных экстремумов многоэкстремальных функций, например корреляционных, используемых при обработке изображений [3]. Одним из эффективных методов решения таких задач является метод мультистарта, широко распространённый на практике в силу простоты его реализации и применимости к любым многоэкстремальным функциям [4, 5].

Метод мультистарта основывается на сведении поиска глобального минимума функции к случайному поиску её локальных минимумов. Для их поиска в данном методе ис-

*Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований (проект № 16-07-00931-а) и Федерального агентства научных организаций (проект № 0348-2017-0010).

пользуются различные алгоритмы спуска, осуществляемого из некоторого множества X, включающего m начальных точек, в локальные минимумы $u_1, u_2, \ldots, u_m$ функции. Наименьшее значение $u^* = \min\limits_{i=\overline{1,m}} u_i$ функции выбирается в качестве её глобального минимума.

В общем виде процесс решения задачи включает три основных этапа:

— построение множества X начальных точек;

— параллельное выполнение спуска из начальных точек для поиска локальных минимумов $u_1, u_2, \ldots, u_m$;

— выбор глобального минимума u^*.

Схема решения задачи представляет собой абстрактную программу, отражающую информационно-логические связи между программными модулями, реализующими её этапы. Данное понятие коррелирует с термином потока заданий workflow, используемым в зарубежных публикациях, например в [6]. Существует широкий спектр систем управления workflow, в их числе известные системы Askalon, Condor DAGMan, Grid Ant, Grid Flow, Karajan, Kepler, Pegasus, Taverna, Triana, UNICORE и другие [7]. Однако анализ возможностей этих систем показывает, что проблемы формирования и обработки ими масштабируемых потоков заданий для центра коллективного пользования не решены в полной мере [8].

В масштабируемом приложении вычислительная нагрузка (поток заданий), связанная с решением задачи, распределяется между элементами ГРВС. В этом случае время выполнения потока заданий уменьшается обратно пропорционально числу используемых элементов с учётом их производительности в составе среды. Таким образом, возникает необходимость исследования входных данных в целях балансировки и максимизации загрузки выделенных ресурсов и достижения тем самым минимизации времени решения задачи.

Рассматриваются вопросы разработки масштабируемого приложения Градиент для решения задач глобальной оптимизации многоэкстремальных функций с помощью инструментального комплекса (ИК) Orlando в ГРВС. Преимущество предложенного подхода заключается в автоматизации балансировки вычислительной нагрузки для элементов среды. Дополнительная возможность приложения состоит в наличии системного модуля для оценки производительности элемента среды при обработке начальных точек. Эффективность приложения показана применительно к многоэкстремальной функции Griewank [9], используемой для тестирования алгоритмов глобальной оптимизации [10].

Постановка задачи. Введём следующие обозначения: k — число вычислительных ядер; π_i — пиковая производительность i-го ядра, представленная отношением числа элементарных операций к единице времени; $\alpha(h)$ — среднее число операций, требуемое для поиска локального минимума некоторой многоэкстремальной функции h из одной начальной точки; d_i — число начальных точек, обрабатываемых i-м ядром; ε_i — накладные расходы i-го ядра, не зависящие от числа начальных точек; τ — время выполнения оптимизационной задачи.

В общем случае в процессе исследования многоэкстремальной функции требуется найти распределение m начальных точек по k группам с числом точек $d_1, d_2, \ldots, d_k$, удовлетворяющее условию

$$\tau = \Big[\max_{i=\overline{1,k}} \Big(\varepsilon_i + \alpha(h) \frac{d_i}{\pi_i} \Big) \Big] \to \min; \qquad \sum_{i=1}^{k} d_i = m. \tag{1}$$

Данная задача является труднорешаемой [11]. В этой связи в предлагаемой работе используется эвристический подход к её решению.

Инструментальный комплекс Orlando. Архитектура комплекса включает следующие основные компоненты: интерфейс пользователя, конструктор модели, базу знаний, исполнительную подсистему и базу расчётных данных.

Интерфейс пользователя обеспечивает ему доступ к компонентам Orlando.

Конструктор модели предназначен для спецификации знаний о программных модулях, схемных знаний о модульной структуре модели и алгоритмов, продукционных знаний для поддержки принятия решений по выбору оптимальных алгоритмов в зависимости от состояния среды, а также знаний о программно-аппаратных характеристиках узлов ГРВС. Эти знания представляются в виде вычислительной модели ГРВС, являющейся частным случаем семантической сети. Модель описывается следующей структурой:

$$M = \langle F, O, Z, N, R_{\text{in}}, R_{\text{out}}, R_{FO}, R_{FN} \rangle,$$

где F, O, Z, N — множества программных модулей, операций, параметров, узлов ГРВС соответственно; $R_{\text{in}} = Z \times O$, $R_{\text{out}} = O \times Z$, $R_{FO} = F \times O$, $R_{FN} = F \times N$ — отношения между элементами множеств F, O, Z и N. Отношения R_{in} и R_{out} типа «многие-ко-многим» задают множества входных и выходных параметров операции и тем самым определяют информационно-логические связи между операциями. Отношение R_{FO} типа «один-ко-многим» устанавливает связи между операциями и реализующими их модулями, а отношение R_{FN} типа «многие-ко-многим» — связи между узлами и модулями, которые могут быть выполнены в узлах. В ГРВС поступает запрос пользователя в непроцедурной форме: вычислить значения параметров из множества $Z_{\text{out}} \subset Z$ по значениям параметров из множества $Z_{\text{in}} \subset Z$. В общем случае в модели РВС может существовать множество S планов решения этой задачи, каждый из которых определяет, какие модули из F и в каком порядке должны быть выполнены.

Основными объектами схемных знаний являются параметры (значимые переменные предметной области) и операции, выполняемые над полем параметров. Модули представляют собой программную реализацию операций. Один модуль может реализовывать несколько операций. Спецификация модуля включает следующую информацию: тип и семантику входных, выходных и транзитных параметров, способы передачи параметров и обработки нестандартных ситуаций, режим запуска и другие сведения.

Средства ИК обеспечивают представление вычислительной модели как в текстовом виде на основе языка разметки Extensible Markup Language (XML), так и с помощью графических структурных схем. Вычислительная модель хранится в базе знаний.

Исполнительная подсистема включает набор интерпретаторов схем решения задач и планировщиков вычислений. Интерпретатор производит обработку управляющих конструкций и выполнение операций схем решения задач. Планировщик осуществляет декомпозицию схем решения задач на подсхемы в целях оптимизации распределения вычислительной и коммуникационной нагрузки на ресурсы ГРВС. Декомпозиция может быть выполнена в статическом режиме до начала вычислений, а также динамически в процессе вычислений.

Исходные данные и результаты вычислений хранятся в базе расчётных данных.

По одному интерпретатору и планировщику размещается во всех управляющих узлах кластеров. В рамках децентрализованной архитектуры исполнительной подсистемы интерпретаторы и планировщики могут поддерживать локальные взаимодействия друг с другом для перераспределения вычислительной нагрузки. В целом исполнительная подсистема обеспечивает реализацию многоуровневого параллелизма в процессе решения задачи.

Масштабируемое приложение Градиент. Разновидности метода мультистарта отличаются алгоритмами выбора начальных точек и поиска локального минимума. Их общим недостатком является многократное нахождение одного и того же локального минимума. Для устранения данной проблемы применяется кластеризация начальных точек.

Однако этот метод также имеет ряд недостатков [4], влияющих на вероятность нахождения глобального минимума. Большинство методов кластеризации требует подбора параметров их работы под конкретную ситуацию, что влечёт за собой необходимость дополнительного исследования функции. Отдельные локальные минимумы не могут быть найдены в случае их расположения в окрестности других локальных минимумов. Некоторые алгоритмы локального спуска из начальных точек, лежащих как угодно близко друг к другу, могут находить локальные минимумы, лежащие на значительном удалении друг от друга. Поэтому в качестве основы выбран алгоритм классического мультистарта.

Приложение Градиент для поиска глобального минимума функции $f(x)$ методом мультистарта включает четыре операции. Параметры и операции пакета Градиент, а также их взаимосвязи представлены на рис. 1. Операция f_1 получает на вход список операций с указанием минимального и максимального числа их экземпляров, порождаемых при ре-

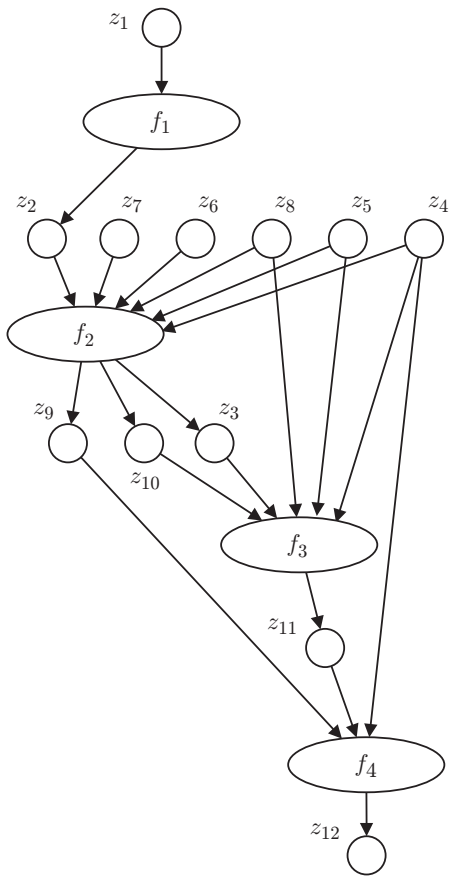

Рис. 1. Схема решения задачи. Параметры: z_1 — число экземпляров операций; z_2 — информация о выделенных ресурсах и их производительности; z_3 — ресурсы, на которых будут запущены экземпляры операций; z_4 — минимизируемая многоэкстремальная функция в текстовом виде; z_5 — размерность функции; z_6 — параметры минимизации; z_7 — область генерации начальных точек; z_8 — число начальных точек; z_9 — число групп с начальными точками; z_{10} — параллельный список групп начальных точек; z_{11} — параллельный список локальных минимумов и их координат; z_{12} — минимальное значение многоэкстремальной функции и координаты точки, в которой оно найдено. Операции: $f_1(z_1 \to z_2)$ — резервирование ресурсов; $f_2(z_2, z_4, z_5, z_6, z_7, z_8 \to z_3, z_9, z_{10})$ — генерация начальных точек; $f_3(z_{10} \to z_{11})$ — осуществление спуска из начальной точки методом градиента; $f_4(z_9, z_{11} \to z_{12})$ — нахождение минимального значения функции

шении задачи. На основе этой информации операция f_1 выдаёт число выделенных ресурсов и их производительность. Данная операция реализуется системным модулем, который осуществляет выделение ресурсов ГРВС и измеряет их производительность с помощью агентов системы метамониторинга [12]. Операция f_1 выполняет дополнительный этап схемы решения задачи по сравнению с классическим методом мультистарта и обеспечивает декомпозицию задачи операцией f_2 на основе эвристической информации о производительности ресурсов.

Операция f_2 генерирует множество начальных точек, которое разбивается на массивы по числу ресурсов. Сопоставление массивов и ресурсов осуществляется с учётом их производительности, что обеспечивает эффективность использования выделенных ресурсов. Операция f_3 реализует спуск из начальной точки в точку локального минимума методом градиента. При выполнении схемы решения задачи интерпретатор запускает множество экземпляров этой операции на выделенных ресурсах. Каждый экземпляр операции реализуется параллельной программой. Элементы параллельных списков z_{10} и z_{11}, являющихся соответственно входным и выходным параметрами операции f_3, обрабатываются независимо друг от друга в отдельных процессах — экземплярах этой операции. Операция f_4 объединяет результаты вычислений и находит минимальное значение функции, принимаемое в качестве её глобального минимума.

Вычислительные эксперименты. Рассмотрим задачу минимизации функции Griewank с глобальным экстремумом в начале координат [9]. Её локальные минимумы находятся в окрестности глобального минимума. Функция имеет следующий вид:

$$f(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad -28 \le x_i \le 28, \quad i = \overline{1,n}.$$

В первом эксперименте исследована масштабируемость вычислений на кластере с использованием от 1 до 30 двухпроцессорных узлов с общим числом ядер узла равным 32. Расчёты проводились при $z_5 = 5000000$. Рис. 2, a, b иллюстрирует изменение времени решения задачи и ускорения вычислений. Результаты получены при следующих значениях n: t_1, t_4, s_1 и s_4 при $n = 2$; t_2, t_5, s_2 и s_5 при $n = 5$; t_3, t_6, s_3 и s_6 при $n = 8$.

Снижение ускорения вычислений связано с ростом доли накладных расходов в общем времени решения задачи. Накладные расходы обусловлены копированием входных данных

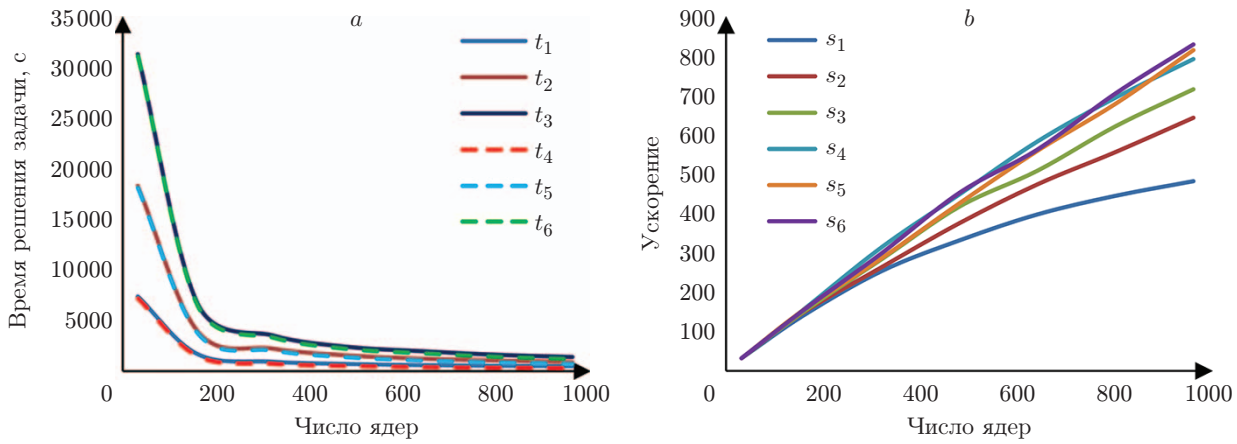

Рис. 2. Показатели процесса решения задачи: a — время решения задачи с учётом (t_1, t_2 и t_3) и без учёта (t_4, t_5 и t_6) накладных расходов; b — ускорение с учётом (s_1, s_2 и s_3) и без учёта (s_4, s_5 и s_6) накладных расходов

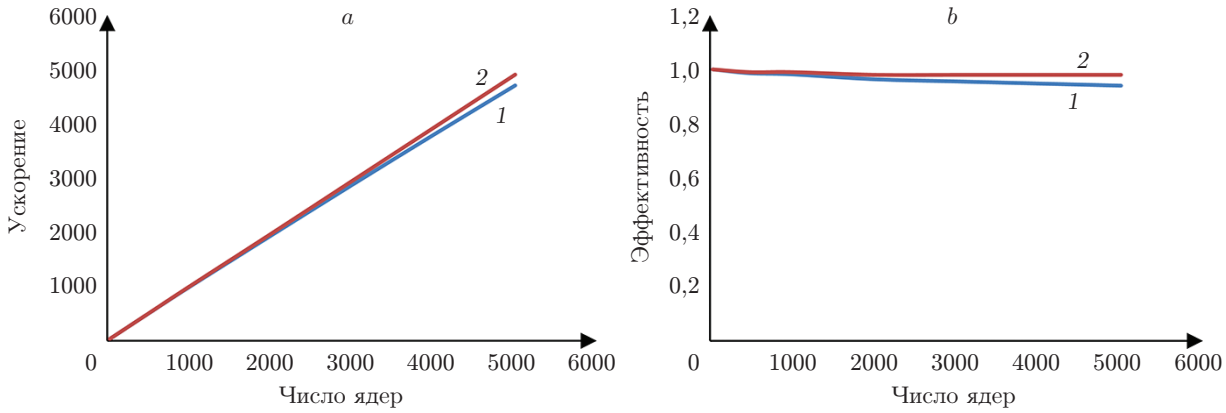

Рис. 3. Прогнозируемые показатели процесса решения задачи: a — ускорение, b — эффективность вычислений (кривые *1* — без учёта, кривые *2* — с учётом накладных расходов)

на кластер, постановкой заданий в очередь, ожиданием запуска заданий, временем опроса состояния заданий, временем копирования результатов и другими действиями.

На рис. 3, a, b приведены прогнозируемые показатели ускорения и эффективности вычислений в процессе оптимизации двухмерной функции Griewank с использованием большего числа вычислительных ресурсов для 500 млн начальных точек. Представленные результаты показывают, что усложнение задачи (числа начальных точек) позволяет повысить эффективность и ускорение вычислений.

Второй эксперимент направлен на исследование различных тактик распределения вычислительной нагрузки в гетерогенной среде. Учёт неоднородности доступных ресурсов при проведении распределённых вычислений является сложной проблемой.

В данной работе применяется метод решения задач прогнозирования времени выполнения программ, базирующийся на использовании специализированных наборов их характеристик и вычислительных узлов, а также специальных математических соотношений между этими характеристиками и показателями времени выполнения исследуемых программ с применением следующих показателей: число инструкций программы; число тактов процессора, необходимых для её выполнения; объёмы данных, записанных/считанных в/из кэш первого и второго уровней; число кэш-промахов первого и второго уровней; число запросов к кэш-памяти; число успешно завершённых запросов и число строк, выделенных в кэш-памяти. В качестве характеристик узла использовались производительность процессора при выполнении целочисленных операций и операций с плавающей точкой, а также размер, пропускная способность и латентность для оперативной и кэш-памяти.

В качестве примера был исследован процесс решения задачи оптимизации двухмерной функции Griewank с 5 млн начальных точек на десяти узлах двух кластеров с различной производительностью (по пять узлов от каждого кластера). Узлы отличались числом процессоров (один или два) и размером оперативной памяти (32 или 64 ГБ). Анализ алгоритма градиентного спуска показал, что размер оперативной памяти при его выполнении не является практически значимым. Число операций с плавающей точкой, выполняемых в секунду, напротив, существенно влияет на время выполнения данного алгоритма. Коэффициенты производительности узлов двух кластеров, полученные в результате проведённого исследования, имеют значения 0,49 и 1 соответственно.

Приведём три тактики распределения вычислительной нагрузки между узлами.

1. Создание для каждого ресурса по одному массиву с равным числом начальных точек.

Показатель	Тактика 1	Тактика 2	Тактика 3
Время, с	1580	1380	1096
Ускорение	4,54	5,20	6,54
Эффективность	0,45	0,52	0,65

2. Формирование избыточного (по отношению к числу ресурсов) числа массивов с равным числом начальных точек. Ресурсы выделяются для выполнения операций с массивами по мере своего освобождения.

3. Назначение каждому ресурсу по одному массиву с числом начальных точек, пропорциональным производительности ресурса для данной задачи. Такая тактика реализована в ИК Orlando в рамках эвристического подхода к решению задачи (1).

Применение различных тактик распределения вычислительной нагрузки между ресурсами свидетельствует о преимуществе третьей тактики распределения вычислительной нагрузки (см. таблицу).

Результаты экспериментов показали хорошую масштабируемость и эффективность вычислений с помощью приложения Градиент при решении задачи поиска глобального минимума многоэкстремальных функций в ГРВС. Все расчёты выполнены с использованием ресурсов Центра коллективного пользования «Иркутский суперкомпьютерный центр СО РАН». Инструментальный комплекс Orlando включён в штатное программное обеспечение этого центра. Показатели ускорения и эффективности в экспериментах вычислены относительно времени решения задач на самом высокопроизводительном узле с использованием 32 ядер.

Заключение. В данной работе предложен новый подход к реализации многоуровневого параллелизма алгоритма решения задачи исследования многоэкстремальных функций с учётом различий характеристик узлов вычислительной среды, измеряемых с помощью агентов системы метамониторинга, а также высокоуровневый программный инструментарий для разработки масштабируемых приложений в ГРВС. Преимущества представленного подхода показаны на примере решения задачи исследования многоэкстремальной функции.

СПИСОК ЛИТЕРАТУРЫ

1. **Georgiou Y., Hautreux M.** Evaluating scalability and efficiency of the Resource and Job Management System on large HPC Clusters // Lecture Notes Comput. Sci. 2013. **7698**. P. 134–156.
2. **Хоменко М. Д., Дубров А. В., Мирзаде Ф. Х.** Стратегии декомпозиции в задачах моделирования процессов аддитивных лазерных технологий // Автометрия. 2016. **52**, № 6. С. 110–119.
3. **Баклицкий В. К., Бочкарев А. М., Мусьяков М. П.** Методы фильтрации сигналов в корреляционно-экстремальных системах навигации. М.: Радио и связь, 1986. 216 с.
4. **Жиглявский А. А., Жилинская А. Г.** Методы поиска глобального экстремума. М.: Наука, 1991. 248 с.
5. **Елесина С. И., Никифоров М. Б.** Исследование особенностей метода мультистарта в глобальной оптимизации // Проектирование и технология электронных средств. 2011. № 2. С. 45–49.
6. **Yu J., Buyya R.** A taxonomy of workflow management systems for grid computing // Journ. Grid Comput. 2005. **3**, N 3–4. P. 171–200.
7. **Talia D.** Workflow systems for science: Concepts and tools // ISRN Software Eng. 2013. **2013**. P. 1–15. URL: https://www.hindawi.com/journals/isrn/2013/404525/ (дата обращения: 9.02.2017).

8. **Tao J., Kolodziej J., Ranjan R. et al.** A note on new trends in data-aware scheduling and resource provisioning in modern HPC systems // Future Generation Comput. Syst. 2015. **51**. P. 45–46.
9. **Cho H., Olivera F., Guikema S. D.** A derivation of the number of minima of the Griewank function // Appl. Math. Computat. 2008. **204**, N 2. P. 694–701.
10. **Jamil M., Yang X.-S.** A literature survey of benchmark functions for global optimisation problems // Intern. Journ. Math. Modelling and Numer. Optimisation. 2013. **4**, N 2. P. 150–194.
11. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
12. **Бычков И. В., Опарин Г. А., Феоктистов А. Г. и др.** Мультиагентное управление вычислительной системой на основе метамониторинга и имитационного моделирования // Автометрия. 2016. **52**, № 2. С. 3–9.

Поступила в редакцию 3 июля 2017 г.