

ОБ ОДНОМ ПОДХОДЕ К ПЛАНИРОВАНИЮ В СЕТИ ДОБРОВОЛЬНЫХ ВЫЧИСЛЕНИЙ

Т. М. Леденева*, С. С. Пархоменко**

*Воронежский государственный университет

**ООО НПП РЕЛЭКС (г. Воронеж)

Поступила в редакцию 20.06.2018 г.

Аннотация. В статье предложена многокритериальная модель назначения работ волонтерам в сети добровольных вычислений, учитывающая фактор неопределенности в определении времени выполнения работ. Решение задачи планирования осуществляется с помощью генетического алгоритма, который позволяет учитывать динамические изменения в вычислительной среде и оперативно на них реагировать.

Ключевые слова: сеть добровольных вычислений, планирование, генетический алгоритм.

Annotation. The article proposes a multi-criteria model of assigning jobs to volunteers in a voluntary network, which takes into account the uncertainty factor in determining the time for performance of work. The solution of the planning task is carried out with the help of a genetic algorithm. It allows to take into account dynamic changes in the computing environment and react to them promptly.

Keywords: voluntary calculation network, planning, genetic algorithm.

ВВЕДЕНИЕ

Добровольные вычисления (Volunteer Computing) – современная информационная технология распределенных вычислений, идея которой заключается в использовании вычислительных ресурсов, добровольно предоставляемых для проведения масштабных расчетов пользователями персональных компьютеров (ПК) – волонтерами, при этом используются только свободные ресурсы, поэтому участие в добровольных вычислениях не мешает работе пользователей. Основным каналом связи выступает сеть Интернет, а ПК волонтеров объединяются в грид под управлением некоторой среды. Такие грид-системы создаются под конкретные масштабные задачи и называются проектами добровольных вычислений (или волонтерскими вычислительными системами). К наиболее известным проектам добровольных вычислений относятся следующие: поиск новых космических объектов (Einstein@home), проверка некоторых гипотез теории чисел (ABC@

home), исследование потоков звездного вещества Млечного пути (MilkyWay@home), компьютерное моделирование свертывания молекул белка (Folding@home) и другие. Из отечественных проектов самый известный – это проект Optima@Home, предназначенный для исследования эффективной реализации методов оптимизации [1]. Наиболее популярная открытая программная платформа для организации добровольных вычислений – это BOINC (Berkeley Open Infrastructure for Network Computing) [2]. Каждый проект добровольных вычислений состоит из трех основных частей: сервера, веб-сайта и прикладного программного обеспечения. На сервере создаются задания для обработки. Волонтер принимает решение об участии в проекте и доле предоставляемых ресурсов. Для того, чтобы подключиться к проекту, волонтер должен установить на свой компьютер специальную программу, которая позволяет подключать ПК к любым проектам, осуществляя обмен данными с сервером выбранного проекта. На рисунке представлена общая модель сети добровольных вычислений BOINC.

© Леденева Т. М., Пархоменко С. С., 2018

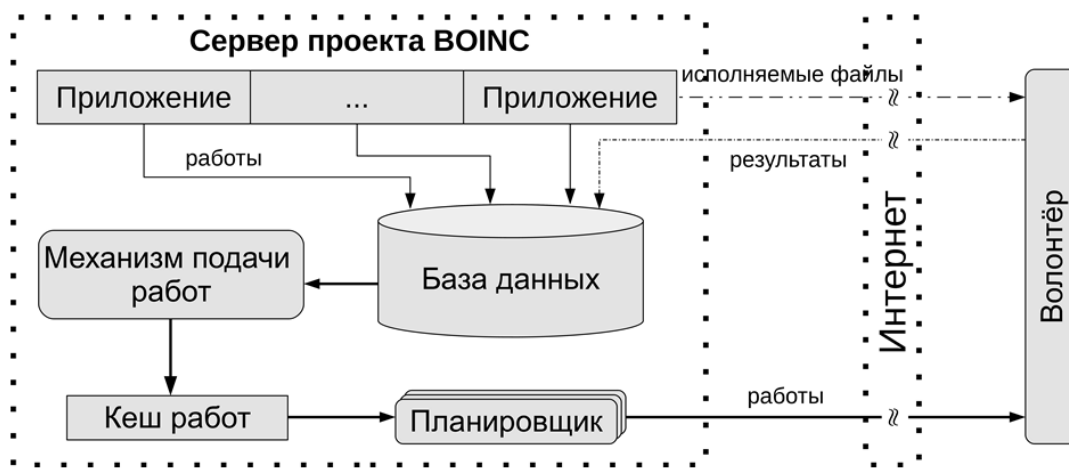


Рис. Общая модель сети добровольных вычислений BOINC

Можно выделить несколько характерных особенностей, присущих сетям добровольных вычислений:

- динамический характер и неравномерность загрузки оборудования;

- большое разнообразие волонтеров подвергает реализацию приложений различным нарушениям;

- выполнение заданий на различных типах ПК требует различного времени;

- на время начала и окончания периодов доступности ресурсов значительное влияние оказывает фактор неопределенности;

- поскольку централизованное управление отсутствует, то успешность вычислительного процесса зависит от волонтеров.

Большое значение для повышения эффективности функционирования сетей добровольных вычислений имеет совершенствование механизмов планирования, основной целью которых является адаптация к волонтерской среде. Развитию методов планирования для сетей добровольных вычислений посвящены работы [3–6]. В [3] вводится понятие периодов доступности ПК волонтеров, к которым и осуществляется «привязка» плана. Большое внимание уделяется разработке стратегий и политик планирования [7–9]. Политики планирования представляют собой набор эвристических правил, которые учитывают такие характеристики, как статистику сетевого подключения волонтера, минимальное время между периодами сетевой активности, характеристики оборудования, сложность заданий и др. Основные алгорит-

мы, которые можно использовать для реализации большинства политик представлены в [8, 9]. Одна из самых существенных проблем – это оценка времени выполнения работ на различных ПК в волонтерской вычислительной системе. В [5, 10] предложен подход к планированию на основе прогнозирования времени выполнения работ с помощью нейронной сети. Данный подход позволяет учитывать все нюансы, влияющие на оценку, в совокупности на исторической базе, которая формируется в процессе работы сети и постоянно обновляется.

Структурно задача планирования относится к классу NP-полных задач дискретной оптимизации, поэтому в общем случае предполагает нахождение удовлетворительного субоптимального решения. Цель статьи заключается в представлении моделей и алгоритмов планирования распределённых вычислений на основе генетического алгоритма с учетом того, что время выполнения работ подвержено фактору неопределенности.

1. МОДЕЛЬ ПЛАНИРОВАНИЯ ДЛЯ СЕТЕЙ ДОБРОВОЛЬНЫХ ВЫЧИСЛЕНИЙ

В широком смысле планирование вычислений – это определение порядка выполнения работ, но при этом предполагается и назначение, т. е. распределение доступных ресурсов между работами.

Введем следующие предположения для разработки моделей планирования, подразумеваемая под исполнителями ПК волонтеров:

а) производительность исполнителей различна, б) исполнители работают параллельно; в) каждая работа неделима и независима, г) процесс выполнения работы непрерывен, в некоторый момент времени исполнитель может выполнять только одну работу, д) идентичные работы относятся к одному типу.

Пусть имеется m волонтеров и p приложений, при этом каждый волонтер $i \in \{1, \dots, m\}$ может выполнить любую работу любого приложения, но для выполнения работы конкретного приложения волонтеры затрачивают разное время. Обозначим через t_{ij} – время выполнения i -м волонтером работы j -го приложения, тогда $T = (t_{ij})_{m \times p}$ – технологическая матрица с элементами t_{ij} . Необходимо распределить n работ, относящихся к различным приложениям, по m волонтерам. Заметим, что сети добровольных вычислений являются динамическими и открытыми, а, следовательно, в процессе их работы допустимы изменения n , m и доли предоставленных ресурсов.

В общем случае оптимизационная задача планирования может быть многокритериальной. Для решения проблемы многокритериальности можно использовать стандартные подходы [11].

Пусть $V = \{1, \dots, m\}$ – множество потенциальных волонтеров; $A = \{1, \dots, p\}$ – множество потенциальных приложений. Каждому j -му приложению поставим в соответствие вес $\mu_j \in [0, 1]$, который отражает приоритетность работ данного приложения, так что $\sum_{j=1}^p \mu_j = 1$. Работы каждого приложения идентичны и требуют для своего выполнения одинаковое время.

Будем считать, что каждый i -й волонтер имеет некоторую репутацию, которая сформировалась на основе опыта выполнения работ на платформе добровольных вычислений и выражается весом $v_i \in [0, 1]$, так что $\sum_{i=1}^m v_i = 1$. Кроме того, у каждого i -го волонтера имеется совокупность периодов доступности $\{[\underline{d}_i^l, \bar{d}_i^l]\}_{l=1, D_i}$ в течение суток, которые необходимо учитывать при планировании.

Каждый i -й волонтер – это ПК некоторого пользователя, который имеет определенную

производительность, оценка которой выражается величиной C_i .

Предположим, что распределение работ осуществляется в некоторый момент времени τ , который инициирует запуск алгоритма планирования.

Введем переменную

$$a_j(\tau) = \begin{cases} 1, & \text{если в момент } \tau \text{ работы } j\text{-го} \\ & \text{приложения включаются в план,} \\ 0, & \text{иначе,} \end{cases}$$

тогда множество $A(\tau) = \{j : a_j(\tau) = 1, j \in A\}$ определяет список приложений, работы которых включаются в план, при этом весовые коэффициенты пересчитываются по формуле

$$\mu'_j = \frac{\mu_j}{\sum_{k \in A(\tau)} \mu_k}.$$

Введем переменную

$$y_{ij} = \begin{cases} 1, & \text{если } \exists l (\tau \in [\underline{d}_i^l, \bar{d}_i^l]) \text{ и } t_{ij} < \bar{d}_i^l - \tau, \\ 0, & \text{иначе.} \end{cases}$$

Данная переменная отражает тот факт, что волонтер находится в сети в момент планирования τ , и он может выполнить работу j -го приложения до окончания его периода доступности. Тогда множество волонтеров, которые могут участвовать в распределении работ в момент времени τ , определяется следующим образом:

$$V(\tau) = \{i : \exists j (y_{ij} = 1), i \in V\},$$

при этом весовые коэффициенты волонтеров пересчитываются по формуле $v'_i = \frac{v_i}{\sum_{k \in V(\tau)} v_k}$.

Пусть c_{ij} – количество работ j -го приложения, назначаемых i -му волонтеру; $n_j(\tau)$ – количество работ j -го приложения, которые необходимо распределить в момент времени τ . Составим ограничения оптимизационной модели с учетом следующих допущений:

1) в момент времени τ известно множество приложений, работы которых необходимо распределять, и множество волонтеров, которые готовы их выполнять, т. е.

$$A(\tau) \subseteq A, V(\tau) \subseteq V; \quad (1)$$

2) количество работ c_{ij} j -го приложения, которые назначаются i -му волонтеру, должно соответствовать его производительности, т. е.

$$f\left(\sum_{j \in A(\tau)} c_{ij}\right) \leq C_{ij} \quad (i \in V(\tau)), \quad (2)$$

где $f(\bullet)$ – оценка производительности, необходимой для выполнения назначенных работ;

3) все работы j -го приложения, доступные для планирования в момент времени τ , должны быть распределены между волонтерами, т. е.

$$\sum_{i \in V_\tau} c_{ij} \cdot y_{ij}(\tau) = n_j(\tau) \quad (j \in A_\tau); \quad (3)$$

4) все работы, доступные для планирования в момент времени τ , должны быть распределены, т. е.

$$\sum_j n_j(\tau) = |A(\tau)|. \quad (4)$$

5) В качестве критериев оптимальности используются следующие целевые функции:

6) минимизация максимального времени волонтеров, предназначенного для добровольных вычислений, т. е.

$$\max_i \left(\sum_{j \in A(\tau)} c_{ij} \cdot y_{ij}(\tau) \cdot t_{ij} \right) \rightarrow \min; \quad (5)$$

7) максимизация суммарного приоритета приложений, работы которых включены в план, т. е.

$$\sum_{j \in A(\tau)} \mu'_j \cdot \sum_{i \in V(\tau)} c_{ij} \cdot y_{ij}(\tau) \rightarrow \max; \quad (6)$$

8) максимизация суммарной репутации волонтеров, которым назначены работы, т. е.

$$\sum_{i=1}^N v'_i \cdot \max_{j=1, P} \{y_{ij}(\tau)\} \rightarrow \max. \quad (7)$$

Помимо перечисленных критериев можно использовать и другие критерии (минимизация максимального времени завершения всех заданий, оптимизация способности плана противостоять нарушениям, максимизация надежности и др.), применяемые при оценке сетей добровольных вычислений [8].

Решение задачи представляется в виде матрицы $C = \{c_{ij}\}_{m \times p}$. Задача планирования решается каждый раз при изменении состава сети. Основная особенность задачи связана с определением технологической матрицы T . Как отмечалось выше, время, необходимое для выполнения работы, может быть различным, при этом желательно учитывать доступ-

ный объем оперативной памяти, скорость процессора, степень требуемой избыточности вычислений для повышения надежности и другие характеристики. Однако более целесообразным является использование некоторых обобщенных показателей, которые способны учитывать и характеристики ПК волонтеров, и сложность решаемых задач, и опыт решения данной задачи на платформе добровольных вычислений. В [5, 10] такие обобщенные показатели формируются на основе нейронной сети в форме интервального прогноза времени выполнения работы. Предположим, что имеется историческая база, на основе которой можно сформировать оценку времени выполнения данной работы. Необходимость учета фактора неопределенности приводит к тому, что помимо числовой информации о времени выполнения задания $t_{ij} \in \mathbb{R}^+$, можно рассматривать следующие варианты приближенной информации:

а) в форме интервального числа $[t_{ij}] = [\underline{t}_{ij}, \bar{t}_{ij}]$;
 б) в форме нечеткого числа \tilde{t}_{ij} с функцией принадлежности $\mu_{\tilde{t}_{ij}}(u)$, где $u \in \mathbb{R}^+ \cup \{0\}$.

Таким образом, целевая функция (4), где оценки времен t_{ij} заданы в форме интервальных или нечетких чисел порождают неординарные задачи планирования, для решения которых необходима разработка специальных подходов. Заметим, что в соответствии с [12] каждое нечеткое число может быть представлено в виде совокупности α -срезов, т. е. интервальных чисел. Каждый α -срез определяется параметром α , который можно интерпретировать, например, как степень доверия к информации. Если такая оценка имеется, то возникает необходимость в разработке подходов к решению интервальной, в которой, с учетом операций над интервальными числами, целевая функция имеет вид

$$\begin{aligned} & \max_i \left(\sum_{j \in A(\tau)} c_{ij} \cdot y_{ij}(\tau) \cdot [t_{ij}] \right) = \\ & = \max_i \left[\sum_{j \in A_\tau} c_{ij} \cdot y_{ij}(\tau) \cdot \underline{t}_{ij}, \sum_{j \in A_\tau} c_{ij} \cdot y_{ij}(\tau) \cdot \bar{t}_{ij} \right] \rightarrow \min. \end{aligned} \quad (9)$$

2. ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПЛАНИРОВАНИЯ

Поскольку задача планирования является NP-полной задачей дискретной оптимизации, то для её решения целесообразно использовать генетический алгоритм, который позволит за приемлемое время найти субоптимальное решение.

Основные понятия, связанные с реализацией генетического алгоритма, можно найти в [13].

Пусть за каждой работой закреплён ген, значение которого отвечает за номер исполнителя. Сформировав и объединив цепи из работ одного типа, получим представление расписания в виде хромосомы длины n вида

$$\mathbf{Ch} = \underbrace{r_1^1 \dots r_{n_1}^1}_{n_1 \text{ работ 1-го приложения}} \underbrace{r_1^2 \dots r_{n_2}^2}_{n_2 \text{ работ 2-го приложения}} \dots \underbrace{r_1^p \dots r_{n_p}^p}_{n_p \text{ работ } p\text{-го приложения}}, \quad (10)$$

где r_k^j (локус хромосомы) – номер исполнителя для k -й работы j -го приложения, $1 \leq r_k^j \leq m$.

Из хромосомы \mathbf{Ch} можно выделить решение в виде матрицы \mathbf{C} с элементами

$$c_{ij} = \sum_{k=1}^{n_j} \phi(i, j, k),$$

$$\phi(i, j, k) = \begin{cases} 1, & \text{если } r_k^j = i, r_k^j \in \mathbf{Ch}, \\ 0, & \text{иначе.} \end{cases}$$

Предложенный генетический алгоритм реализован на основе классической схемы и имеет следующие особенности:

1) В качестве функции приспособленности используется целевая функция (9).

2) Отбор (селекция) производится следующим образом: популяция выстраивается в линейный список, и каждая особь на нечётной позиции скрещивается с соседом, стоящим впереди. Такой подход позволяет реализовать частично параллельную версию генетического алгоритма, гарантируя участие всех особей без дополнительных механизмов контроля за использованием хромосомы в вычислениях (синхронизации).

3) В качестве оператора скрещивания используется модифицированный односточный кроссинговер. Первая модификация включает в себя запрет на скрещивание

между родственниками, под которыми подразумеваются особи, представляющие одинаковое решение с помощью различных хромосом. Вторая модификация заключается в локальном применении селекции. Из четвёрки особей места родителей займут те, у кого значения целевой функции меньше, чем у остальных.

4) Поскольку для селекции популяция выстроена в линейный список, то для того, чтобы особь имела возможность скрещиваться с другими членами популяции, используется случайная мутация.

5) Время работы генетического алгоритма ограничено числом итераций.

Сеть добровольных вычислений является динамичной, в процессе её функционирования изменяется состав волонтеров и множество распределяемых работ. Эти изменения отражаются в популяции, которая модифицируется после каждой итерации генетического алгоритма. Варьируя диапазон допустимых значений r_k^j для (10), можно добавлять и исключать волонтеров, участвующих в планировании, а с помощью перераспределения локусов в хромосоме, можно изменять состав выбранных работ. Такие модификации позволяют реагировать на изменения в вычислительной среде до момента отправки готовых планов волонтерам без полной потери найденного субоптимального решения. Для переориентации направления поиска временно увеличивается активность оператора мутации. Предполагается, что планирование осуществляется циклично на основе динамично обновляемых расписаний, формируемых в моменты поступления нового задания в систему. Субоптимальное планирование позволяет обеспечить высокую скорость работы вычислительной сети.

3. ИЛЛЮСТРАТИВНЫЙ ПРИМЕР

Рассмотрим работу генетического алгоритма на примере. Пусть необходимо распределить $n_1 = n_2 = n_3 = n_4 = 5$ работ (всего 20) по $m = 3$ волонтерам при условии, что для каждого волонтера определено прогнозируемое время выполнения каждого вида работ в форме ин-

тервального числа, и технологическая матрица имеет вид

$$\tilde{T} = \begin{pmatrix} [3,10] & [15,18] & [25,30] & [30,33] \\ [7,10] & [10,20] & [27,33] & [30,32] \\ [1,10] & [11,25] & [25,32] & [23,33] \end{pmatrix}.$$

Предположим, что в момент времени τ генетический алгоритм сформировал следующую популяцию

$$Ch_1 = 23223 \ 12221 \ 11121 \ 33332,$$

$$Ch_2 = 13131 \ 12221 \ 1113 \ 22332,$$

$$Ch_3 = 13131 \ 12221 \ 11231 \ 23233,$$

$$Ch_4 = 31131 \ 12221 \ 11221 \ 33332,$$

$$Ch_5 = 11213 \ 21222 \ 23213 \ 31112,$$

$$Ch_6 = 23131 \ 12221 \ 11132 \ 23233,$$

$$Ch_7 = 22132 \ 21211 \ 11213 \ 23332.$$

Составим для Ch_1 соответствующую матрицу распределения работ C_1 . Выделив в из Ch_1 все работы приложения $j=1$, получим набор генов 23223. Найдем работы, назначенные волонтерам:

$$i=1 \Rightarrow 23223 \Rightarrow k = \emptyset,$$

$$i=2 \Rightarrow \mathbf{23223} \Rightarrow k = \{1,3,4\},$$

$$i=3 \Rightarrow \mathbf{23223} \Rightarrow k = \{2,5\}.$$

Найдем

$$c_{11} = \phi(1,1,1) + \phi(1,1,2) + \phi(1,1,3) + \phi(1,1,4) + \phi(1,1,5) = 0 + 0 + 0 + 0 + 0 = 0,$$

$$c_{21} = \phi(2,1,1) + \phi(2,1,2) + \phi(2,1,3) + \phi(2,1,4) + \phi(2,1,5) = 1 + 0 + 1 + 1 + 0 = 3,$$

$$c_{31} = \phi(3,1,1) + \phi(3,1,2) + \phi(3,1,3) + \phi(3,1,4) + \phi(3,1,5) = 0 + 1 + 0 + 0 + 1 = 2.$$

Вычислив остальные элементы, получим следующую матрицу распределения работ между волонтерами:

$$C_1 = \begin{pmatrix} 0 & 2 & 4 & 0 \\ 3 & 3 & 1 & 1 \\ 2 & 0 & 0 & 4 \end{pmatrix}.$$

Из нее следует, что первому волонтеру назначаются 3 работы первого типа и 2 работы третьего типа. Теперь определим время обработки работ волонтерами и найдем значение целевой функции

$$f_1 = [0,0] + 2 \cdot [15,18] + 4 \cdot [25,30] + [0,0] = [130,156],$$

$$f_2 = 3 \cdot [7,10] + 3 \cdot [10,20] + [27,33] + [30,32] = [108,155],$$

$$f_3 = 2 \cdot [1,10] + [0,0] + [0,0] + 4 \cdot [23,33] = [94,152].$$

$$\text{Тогда } R(Ch_1) = \max_i \{f_i\} = \max \{[130,156], [108,155], [94,152]\} = [130,156].$$

Аналогичным образом вычисляя оценки остальных хромосом, получим

$$R(Ch_2) = [120,156], \quad R(Ch_3) = [117,157],$$

$$R(Ch_4) = [114,158], \quad R(Ch_5) = [139,188],$$

$$R(Ch_6) = [124,167], \quad R(Ch_7) = [128,167].$$

Ориентируясь на оптимизацию пессимистичного сценария выполнения работ, будем использовать правило сравнения двух хромосом Ch_i и Ch_j вида

$$R(Ch_i) \succ R(Ch_j) \Leftrightarrow$$

$$\Leftrightarrow (\overline{Ch_i} > \overline{Ch_j}) \vee (\overline{Ch_i} = \overline{Ch_j}) \wedge (\underline{Ch_i} > \underline{Ch_j}).$$

$$R(Ch_i) \succ R(Ch_j) \Leftrightarrow$$

$$\Leftrightarrow (\overline{Ch_i} > \overline{Ch_j}) \vee ((\overline{Ch_i} = \overline{Ch_j}) \wedge (\underline{Ch_i} > \underline{Ch_j})).$$

Тогда получим, что оптимальным планом-хромосомой является Ch_2 , поскольку $Ch_3 - Ch_7$ отсеиваются из-за завышенной верхней оценки, а Ch_1 - из-за нижней.

Заметим, что генетический алгоритм даже в простейшей реализации справляется с важнейшей проблемой добровольных вычислений - изменением состава сети. Кроме того, он допускает использование приближенной информации о времени выполнения работ в форме интервальных чисел, а, следовательно, и нечетких чисел, поскольку их можно представить в виде совокупности интервальных.

ЗАКЛЮЧЕНИЕ

Повышение эффективности сетей добровольных вычислений невозможно без совершенствования механизмов планирования. Сети добровольных вычислений пред-

ставляют собой постоянно динамичную вычислительную систему, в большой степени подверженную влиянию факторов неопределенности, что требует адекватного представления информации. Предложена модель планирования работ в форме многокритериальной оптимизационной задачи, а также реализация генетического алгоритма для нахождения субоптимального решения. Генетический алгоритм позволяет справиться с неопределенностью в поведении волонтеров. Планируется, что дальнейшие исследования будут связаны с многовариантными оптимизационными моделями планирования и формализацией различных стратегий и политик в таких моделях.

СПИСОК ЛИТЕРАТУРЫ

1. Заикин, О. С. Опыт организации добровольных вычислений на примере проектов OPTIMA@home и SAT@home / О. С. Заикин, М. А. Посыпкин, А. А. Семенов, Н. П. Храпов // Вестник Нижегородского университета им. Н.И. Лобачевского. Сер. Информационные технологии, 2012. – № 5(2). – С. 340–347.
2. Anderson, D. P. Celebrating diversity in volunteer computing / D.P. Anderson, K. Reed // Proceedings of the 42nd Annual Hawai'i International Conference on System Sciences. – Wai-koloa, 2009. – P. 1–8.
3. Essa, A. An Efficient Algorithm for Scheduling Jobs in Volunteer Computing Platforms / D. Trustram, Z. Zaidi // IEEE International Parallel Distributed Processing Symposium Workshops, 2014. – Pp. 68-76.
4. Ujhelyi, M. Task Scheduling in Distributed Volunteer Computing Systems / M. Ujhelyi, P. Lacko, A. Paulovi // IEEE 12th International Symposium on Intelligent Systems and Informatics, 2014. – P. 111–114.
5. Parkhomenko, S. S. Scheduling in Volunteer networks, based on neural network prediction of the job execution time / S. S. Parkhomenko, T. M. Ledeneva // International Journal of Parallel, Emergent and Distributed Systems. – <https://doi.org/10.1080/17445760.2018.1496435>.
6. Пархоменко, С. С. Составление расписания для распределённых вычислительных сетей в условиях неточных данных / С. С. Пархоменко, Т. М. Леденева // Междунар. науч.-исслед. журн.: Сб. результатов XLII заоч. науч. конф. Research Journal of International Studies. Екатеринбург: МНИЖ – 2015. – № 8. – Ч. 2. – С. 68–72.
7. Топорков, В. В. Стратегии планирования вычислений в распределенных средах / В. В. Топорков // Вестник МЭИ, 2008. – № 5. – С. 33–46.
8. Шамакина, А. В. Обзор технологий распределенных вычислений / А. В. Шамакина // Вестник ЮУрГУ. Сер. Вычислительная математика и информатика, 2014. – Т. 3, №3. – С. 51–85.
9. Топорков, В. В. Модели распределенных вычислений / В. В. Топорков. – М. : ФИЗМАТЛИТ, 2004. – 320 с.
10. Пархоменко, С. С. Обучение нейронных сетей методом Левенберга-Марквардта в условия большого количества данных / С. С. Пархоменко, Т. М. Леденева // Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – 2014. – № 2. – С. 98–106.
11. Леденева, Т. М. Модели и методы принятия решений / Т. М. Леденева. – Воронеж : ВГТУ, 2004. – 189 с.
12. Кофман, А. Введение в теорию нечетких множеств / А. Кофман. – М. : Радио и связь, 1982. – 432 с.
13. Гладков, В. А. Генетические алгоритмы / В. А. Гладков, В.В. Курейчик, В. М. Курейчик. – М. : Физматлит, 2010. – 368 с.

Леденёва Татьяна Михайловна – д-р техн. наук, проф., заведующий кафедрой вычислительной математики и прикладных информационных технологий Воронежского государственного университета.

Тел.: +7(473) 228-11-60 (доб. 1470).

E-mail: ledeneva-tm@yandex.ru

Ledeneva Tatyna Michaylovna – Doctor of Technic Sciences, Professor, Head of the Department of Computational Mathematics and Applied Information Technology, Voronezh State University.

Tel.: +7(473) 228-11-60 (1470).

E-mail: ledeneva-tm@yandex.ru

Пархоменко Станислав Сергеевич – канд. техн. наук, старший инженер, ООО НПП РЕЛЭКС.

E-mail: ibm8086@yandex.ru

Parkhomenko Stanislav Sergeevich – Candidate of Technical Sciences, senior engineer, Research and Production Enterprise RELEX.

E-mail: ibm8086@yandex.ru