УДК 519.8

ЗАДАЧА ПОСТРОЕНИЯ РАСПИСАНИЯ КОНФИГУРАЦИЙ ДЛЯ БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЕЙ

Е.А. Наградов

В целях максимизации продолжительности функционирования сети до исчерпания запаса энергии первого из узлов предложен централизованный алгоритм формирования расписания, основанный на построении набора конфигураций с помощью алгоритма Гарга—Конеманна и последующем решении задачи линейного программирования над построенными конфигурациями.

Ключевые слова: сенсорная сеть, маршрутизация, расписание.

ВВЕДЕНИЕ

Беспроводные сенсорные сети получают широкое распространение при решении задач сбора информации о состоянии окружающей среды, наблюдения за состоянием вулканов [1], обнаружения лесных пожаров [2], наблюдения за состоянием ледников [3].

Для обеспечения автономности узлов, как правило (см., например, работы [1, 2, 4]), в качестве источника энергии узла используется батарея, в связи с чем возникает задача обеспечения максимальной продолжительности функционирования узлов сети.

Будем рассматривать сенсорные сети, для которых характерны следующие особенности.

- В сети явно выделен один узел — базовая станция, через которую производится передача данных из сети. Все узлы сети, за исключением базовой станции, обладают ограниченным запасом энергии.
- Узлы сети неподвижны, в результате чего множество соседних узлов (т. е. узлов, которые могут получать сообщения от данного узла или передавать сообщения данному узлу) для каждого из узлов сети не изменяется в процессе функционирования сети.
- Сценарий использования сети заключается в сборе сообщений от датчиков, содержащихся в узлах сети, и передаче их на базовую станцию. При этом инициаторами передачи данных выступают узлы сети.

Для сетей рассматриваемого класса будем решать задачу максимизации продолжительности функционирования сети до отказа первого из узлов сети из-за исчерпания ресурса батареи.

1. КОНФИГУРАЦИЯ СЕТИ

В настоящей работе рассматривается способ максимизации продолжительности жизни сети, основанный на разделении ролей узлов. Далее предполагается использование в сети протокола MAC-уровня, определяющего следующие роли (режимы функционирования) узлов: маршрутизатор и листовой узел.

В задачи листовых узлов входит сбор данных с датчиков и, в случае появления необходимых для передачи сообщений, передача сообщений маршрутизатору. На маршрутизаторы дополнительно возлагается задача ожидания сообщений от соседних узлов (как маршрутизаторов, так и листовых узлов) и передача полученных сообщений в направлении базовой станции. В отличие от маршрутизаторов, для листовых узлов не требуется выполнять прослушивание радиоканала, в результате чего их энергопотребление существенно ниже (1:3—1:8 по отношению к энергопотреблению уз-

ла-маршрутизатора в зависимости от пропускной способности сети [5, 6]). В работе [5] предложена модификация стека протоколов ZigBee, позволяющая отключать радиопередатчик листовых узлов при отсутствии готовых для передачи сообщений.

Для обеспечения возможности передачи данных от каждого из узлов до базовой станции формируется «конфигурация» сети, определяющая распределение ролей узлов в сети и направления передачи сообщений от узлов до базовой станции. Конфигурация сети является остовным деревом с корнем в базовой станции. Нелистовые узлы конфигурации функционируют в роли маршрутизаторов, связи между узлами определяют направления передачи сообщений.

Далее затраты энергии узлов на передачу сообщений в сети считаются пренебрежимо малыми по сравнению с затратами энергии на ожидание сообщений. Таким образом, энергопотребление узлов определяется только ролью узлов в текущей конфигурации сети.

В связи с тем, что потребление энергии маршрутизаторов превосходит энергопотребление листовых узлов, для увеличения продолжительности функционирования узлов требуется перераспределять роли узлов, изменяя конфигурацию сети.

1.1. Централизованное управление конфигурацией сети

В настоящей работе используется централизованный подход к формированию конфигураций сети. Общий принцип централизованного управления конфигурацией сети заключается в следующем.

1. На начальном этапе функционирования сети каждый из узлов сети собирает информацию о том, какие узлы являются для него соседними.

2. Каждый из узлов передает базовой станции информацию о запасе энергии и перечень соседних с ним узлов.

3. Базовая станция на основании информации о наборе соседних узлов формирует граф сети. На основании графа сети и информации о запасе энергии узлов базовая станция вычисляет динамику изменения конфигураций для сети (расписание). Расписание определяет набор конфигураций сети и продолжительность использования каждой из конфигураций.

4. Базовая станция выбирает первую конфигурацию из расписания.

5. Базовая станция передает каждому из узлов сети информацию о роли узла в новой конфигурации и идентификатор родительского узла.

6. По окончании процедуры распространения узлам информации о конфигурации, базовая станция инициирует смену конфигурации посредством передачи всем узлам соответствующей команды.

7. По истечении продолжительности использования конфигурации, определяемой расписанием, базовая станция выбирает следующую конфигурацию из расписания и повторяет пп. 5 и 6.

Применимость централизованного управления для сетей стандарта ZigBee [7] показана в работе [5], где предложен жадный алгоритм формирования конфигураций сети для случая одинакового начального запаса энергии узлов, основанный на раскраске графа связности. В данной работе предлагается альтернативный способ решения задачи, основанный на сведении задачи к задаче непрерывного линейного программирования, допускающий применение при различном начальном запасе энергии.

2. ЗАДАЧА ПОСТРОЕНИЯ РАСПИСАНИЯ КОНФИГУРАЦИЙ

Приведем формальную постановку задачи построения расписания конфигураций сети для рассматриваемого класса сенсорных сетей.

Множество узлов сети. Обозначим $V = \{v_i\}_{i=0}^{n}$ — множество узлов сети. Узел v_0 является базовой станцией сети.

Домены узлов сети. На начальном этапе функционирования сети узлы собирают информацию о собственном домене (множестве узлов, сообщения от которых узел может принимать). Обозначим $D(v) \subseteq V$ — домен узла v.

Граф сети. На основании информации о доменах узлов сети, базовая станция формирует «граф сети». Граф сети — неориентированный граф (V, E), где V — множество вершин, совпадающее с множеством узлов сети, а $E \subseteq V \times V$ — множество ребер. Ребро между узлами v_i и v_j существует тогда, когда узлы входят в домены друг друга: $(v_i, v_j) \in E \Leftrightarrow (v_i \in D_j) \wedge (v_j \in D_i)$.

Конфигурация сети. Для заданного графа сети (V, E) конфигурацией сети будем называть остовное дерево $q = (V, \hat{E})$ с корнем в вершине v_0. Обозначим $V_R \in V$ — множество узлов, функционирующих в роли маршрутизаторов, $V_S = V \backslash V_R$ — множество листовых узлов. Множества V_R и V_S для заданной конфигурации $q = (V, \hat{E})$ могут быть определены следующим образом: $V_R = \{v_i | d(v_i) > 1\} \cup v_0$,

$V_S = V \setminus V_R$, где $d(v)$ — степень вершины v в дереве $(V, \hat{E})$.

Энергопотребление узлов сети. Для каждого из узлов сети $v \in V$ определим энергопотребление в конфигурации $q = (V, \hat{E})$ в единицу времени следующим образом: $e(q, v) = e_r$, если v — маршрутизатор в конфигурации q, $e(q, v) = e_s$, если v — листовой узел.

Расписание конфигураций. Расписанием конфигураций будем называть множество пар $S = \{(q_i, t_j)\}_{j=1}^{m}$, где $q_j = (V, \hat{E})$ — конфигурация сети, t_j — продолжительность использования конфигурации. Продолжительностью использования расписания будем называть $t(S) = \sum_{j=1}^{m} t_j$. Расписание S будем называть корректным, если до момента времени $t(S)$ ни один из узлов сети не израсходует начальный запас энергии.

Сформулируем оптимизационную задачу построения расписания конфигураций:

Задача 1. *Для заданного графа сети (V, E), начального запаса энергии узлов $b(v)$, энергопотребления маршрутизаторов в единицу времени e_r, энергопотребления листовых узлов в единицу времени e_s требуется построить корректное расписание конфигураций сети максимальной продолжительности.* ♦

3. АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ

3.1. Сведение исходной задачи к задаче непрерывного линейного программирования

Обозначим $Q = \{q_j\}_{j=1}^{m}$ — множество всех возможных конфигураций для графа сети (V, E). Каждой конфигурации $q_j \in Q$ сопоставим продолжительность t_j ее использования. Тогда задача 1 может быть сформулирована следующим образом:

максимизировать

$$\sum_{j=1}^{m} t_j \tag{1}$$

при условиях

$$\sum_{j=1}^{m} e(q_j, v_i) t_j \leqslant b(v_i), \quad i = 1, \ldots, n, \tag{2}$$

$$t_j \geqslant 0. \tag{3}$$

Задача (1)—(3) является задачей непрерывного линейного программирования. Условие (2) представляет собой ограничение корректности расписания. При заданном множестве Q точное решение задачи может быть получено посредством симплекс-метода.

3.2. Описание алгоритма решения задачи

В связи с тем, что множество Q всех возможных конфигураций сети экспоненциально [8] зависит от числа узлов сети, вычисление всего множества конфигураций перед решением задачи непрерывного линейного программирования является вычислительно сложным. Поэтому в данной работе предлагается построить подмножество конфигураций $Q' \subset Q$, для которого искомое расписание может быть получено решением задачи линейного программирования (1)—(3):

- Сформируем подмножество конфигураций $Q' \subset Q$ (см. пп. 3.2.1). Множество Q' строится таким образом, чтобы среди элементов множества были конфигурации, максимально различающиеся набором маршрутизаторов.
- На основе множества Q' сформируем набор условий для задачи (1)—(3) и решим задачу с помощью симплекс-метода (см. пп. 3.2.3). Искомое расписание получим следующим образом: $S = \{(q_j, t_j) | t_j > 0\}$.

В § 4 приводится сравнение эффективности предложенного алгоритма и алгоритма, основанного на способе построения множества Q', приведенного в работе [5].

3.2.1. Формирование множества Q'

Для построения множества конфигураций используется эвристический алгоритм, основанный на алгоритме Гарга—Конеманна — многошаговом эвристическом алгоритме для решения задачи непрерывного линейного программирования (1)—(3) без первоначального построения множества конфигураций Q.

Основная идея алгоритма состоит в следующем.

1. Каждому из узлов сети v_i, $i = 1, \ldots, n$, присваивается вес $y(v_i)$.

2. На каждом шаге алгоритма для заданных весов узлов выполняется построение конфигурации q с минимальным значением стоимости $\omega = \sum_{i=1}^{n} e(q, v_i) y(v_i)$. Описание эвристического алгоритма для построения такой конфигурации приведено далее в пп. 3.2.2. Построенная конфигурация добавляется в расписание.

3. По окончании шага вес узлов увеличивается на величину, пропорциональную их потреблению энергии в построенной конфигурации и заданному параметру ε. Таким образом, на следующем шаге алгоритма, конфигурации, повторно использующие уже выбранные ранее узлы-маршрутизаторы, будут обладать большей стоимостью.

В результате работы алгоритма Гарга—Конеманна получаем множество пар $(q_j, t_j)_{j=1}^{l}$, где q_j — построенные конфигурации, а t_j — продолжительность использования конфигурации. Параметр ε определяет точность алгоритма и число шагов алгоритма, которое для заданного значения ε не превосходит $\left[\frac{1}{\varepsilon}\log_{1+\varepsilon}(n)\right]$ [9].

Заметим, что алгоритм Гарга—Конеманна может применяться и без последующего повторного решения задачи непрерывного линейного программирования (1)—(3), однако, как будет показано далее, такой способ решения задачи обладает меньшей эффективностью (см. пп. 3.2.3 и п. 4.1).

3.2.2. Задача построения конфигурации минимальной стоимости

В контексте решаемой задачи 1 сформулируем подзадачу построения конфигурации сети с минимальной стоимостью для заданных весов узлов, которая решается на каждом шаге алгоритма Гарга—Конеманна:

Задача 2. *Для заданных графа сети* (V, E), *потребления энергии в единицу времени для маршрутизаторов и листовых узлов* e_r *и* e_s *соответственно и весов узлов* $y(v_i)$ *требуется построить конфигурацию сети* $q = (V, \widehat{E})$, *минимизирующую стоимость* $\omega = \sum_{i=1}^{n} e(q, v_i)y(v_i)$, *где* $e(q, v_i)$ *— энергопотребление узла* v_i *в конфигурации* q. ♦

Для решения задачи 2 в данной работе применяется жадный эвристический алгоритм, состоящий из следующих шагов.

Шаг 1. Все узлы, за исключением узла v_0, окрашиваются белым цветом. Узел v_0 окрашен серым цветом. Множество ребер в дереве конфигурации пусто.

Шаг 2. Среди узлов серого цвета выбирается такой узел с максимальным значением $|C_0(v)|/y(v)$, где $C_0(v)$ — множество узлов, соседних с узлом v, окрашенных белым цветом. Узел v помечается черным цветом, и для каждого соседнего с ним белого узла v' выполняется следующее: узел v' окрашивается серым, и в множество ребер добавляется ребро (v, v').

Шаг 3. Если не осталось ни одного узла, окрашенного белым, то останов. Иначе переход на шаг 2.

Алгоритм позволяет построить конфигурацию в том случае, если граф сети связный. Результат работы алгоритма — конфигурация $q = (V, \widehat{E})$, множество маршрутизаторов которой соответствует узлам, окрашенным черным цветом.

Теорема 1. *Если для алгоритма решения задачи поиска конфигурации минимальной стоимости (задача 2) выполняется оценка точности* $\omega \leqslant f\omega_{\text{opt}}$, *где* ω_{opt} *— значение стоимости для оптимального решения, то для решения задачи 1 алгоритмом Гарга—Конеманна выполняется оценка точности* $t_{\text{opt}}/t \leqslant (1-\varepsilon)^{-2}f$, *где* t *— продолжительность расписания, полученного алгоритмом,* t_{opt} *— максимальная продолжительность расписания,* $\varepsilon \in (0, 1)$ *— параметр алгоритма Гарга—Конеманна.* ♦

Доказательство приводится в работе [9].

Теорема 2. *Для предлагаемого в данной работе алгоритма решения задачи 2 при условии* $e_r > e_s$ *выполняется оценка точности* $\omega \leqslant \omega_{\text{opt}}e_r/e_s$. ♦

В самом деле, пусть ω_{opt} — точное решение задачи 2. Тогда $\omega_{\text{opt}} = e_r \sum_{v_i \in V_{R\text{opt}}} y(v_i) + e_s \sum_{v_i \in V_{S\text{opt}}} y(v_i) \geqslant$

$\geqslant e_s \sum_{i=1}^{n} y(v_i)$, $\omega = e_r \sum_{v_i \in V_R} y(v_i) + e_s \sum_{v_i \in V_S} y(v_i) \leqslant$

$\leqslant e_r \sum_{i=1}^{n} y(v_i)$, $\frac{\omega}{\omega_{\text{opt}}} \leqslant \frac{e_r \sum_{i=1}^{n} y(v_i)}{e_s \sum_{i=1}^{n} y(v_i)} = \frac{e_r}{e_s}$.

Теорема 3. *Для предложенного алгоритма решения задачи 1 на основе алгоритма Гарга—Конеманна выполняется оценка точности* $t_{\text{opt}}/t \leqslant (1-\varepsilon)^{-2}e_r/e_s$, *где* $\varepsilon \in (0, 1)$ *— параметр алгоритма Гарга—Конеманна.* ♦

Доказательство следует из теорем 1 и 2.

3.2.3. Формирование расписания на основе множества Q'

Особенность решения задачи (1)—(3), получаемого непосредственно из алгоритма Гарга—Конеманна, заключается в чрезвычайно большом числе конфигураций в получаемом расписании (см. далее п. 4.1), в результате чего построенное расписание не применимо для централизованного управления конфигурацией сети в связи с существенными затратами на формирование конфигурации.

Поэтому предлагается выполнить пост-обработку решения, полученного на первом шаге, в целях сокращения числа конфигураций в построенном расписании.

На основании множества Q', полученного на первом шаге, сформируем набор условий для задачи непрерывного линейного программировании (1)—(3) согласно описанию, приведенному в п. 3.1. Для решения построенной задачи был применен симплекс-метод. Исследуем эффективность этого шага алгоритма (см. п. 4.1).

4. ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРЕДЛОЖЕННЫХ АЛГОРИТМОВ

4.1. Исследование эффективности шага пост-обработки решения алгоритма Гарга—Конеманна

Для экспериментов были выбраны сценарии, представленные в табл. 1.

Таблица 1

Сценарии функционирования сети

Сценарий	Число узлов	Максимальный радиус
1	50	30
2		50
3	80	30
4		50

Таблица 2

Эффективность применения шага пост-обработки

Сценарий	ε	$\|S(GK)\|$	$t(GK)$	$\|S(LP)\|$	$t(LP)$
1	0,5	24	112,6	4	166,6
	0,3	73	139,5	6	
	0,1	371	157,9	5	
2	0,5	35	164,3	21	336,7
	0,3	121	237,5	26	357,1
	0,1	634	316,0	35	
3	0,5	36	152,4	18	277,7
	0,3	127	215,2	16	
	0,1	1184	262,6	20	
4	0,5	45	190,5	13	382,3
	0,3	159	269,4		
	0,1	1510	345,4		

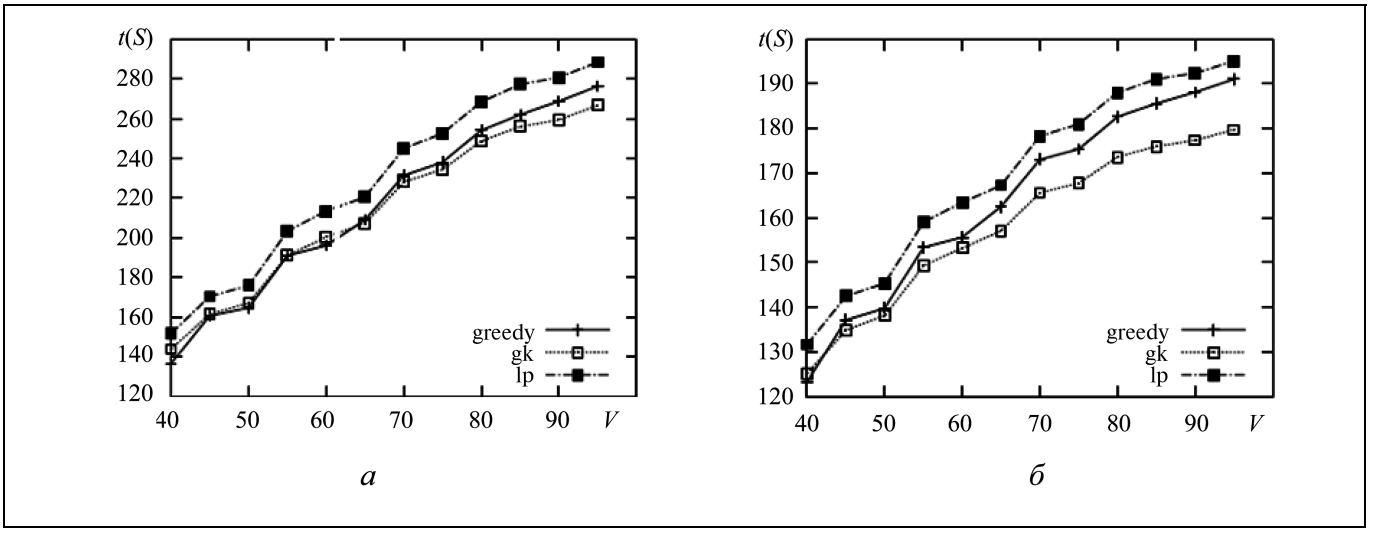
Граф сети формировался следующим образом: заданное число узлов случайным образом распределялось внутри прямоугольной области размером 100 × 100; между узлами существует ребро в графе сети, если расстояние между ними не превосходит заданного максимального радиуса передачи.

Для решения задачи линейного программирования (1)—(3) с помощью симплекс-метода применялась библиотека LPSOLVE [10]. В табл. 2 приведены характеристики числа конфигураций в расписании и продолжительности расписания для схемы с использованием пост-обработки (*LP*) и без (*GK*) в зависимости от выбора параметра ε алгоритма Гарга—Конеманна. Результаты, приведенные в таблице, показывают, что применение предложенного метода пост-обработки решения позволяет не только существенно сократить число конфигураций в получаемом расписании, но и в ряде случаев (в зависимости от значения ε) существенно повысить продолжительность расписания по сравнению с решением алгоритма Гарга—Конеманна.

4.2. Сравнение эффективности алгоритмов решения задачи

Рассмотрим результаты сравнения эффективности предлагаемого алгоритма и алгоритма из работы [5], основанного на построении множества конфигураций с непересекающимися наборами маршрутизаторов. Непосредственное применение алгоритма без модификации не представляется возможным, поскольку алгоритм не учитывает расход энергии нелистовых узлов. Таким образом, для исследования эффективности алгоритм из работы [5] был применен для построения множества конфигураций Q' с последующим формированием на основании построенного множества задачи непрерывного программирования (1)—(3) и решения ее симплекс-методом.

На рисунке приведены графики зависимости средней продолжительности построенных расписаний от числа узлов сети для различных значений e_r и e_s. Усреднение проводилось по 10-ти испытаниям. Меткой «greedy» обозначен модифицированный алгоритм, построенный на основании работы [5], «gk» обозначает алгоритм Гарга—Конеманна без последующей пост-обработки для ε = 0,1, «lp» обозначает предлагаемый алгоритм (схему с последующей пост-обработкой решения алгоритма Гарга—Конеманна при ε = 0,1). Узлы расположены в прямоугольной области размером 100 × 100 с максимальным радиусом передачи 30. Видно, что предложенный алгоритм обладает более высокой



Среднее значение продолжительности расписания в зависимости от числа узлов при начальном запасе энергии $b(v) = 100$ и характеристиках потребления энергии $e_r = 1{,}0$, $e_s = 0{,}2$ (*а*) и $e_r = 1{,}0$, $e_s = 0{,}4$ (*б*)

эффективностью (по критерию продолжительности построенного расписания), чем алгоритм из работы [5].

ЗАКЛЮЧЕНИЕ

Предложен централизованный алгоритм формирования расписания конфигураций для беспроводной сенсорной сети, применимый для случая различного начального запаса энергии узлов и обладающий более высокой эффективностью (по критерию продолжительности построенного расписания), по сравнению с известным алгоритмом. Пост-обработка решения алгоритма Гарга—Конеманна позволила существенно сократить число конфигураций в расписании и таким образом повысить применимость алгоритма для централизованного управления конфигурацией сети.

ЛИТЕРАТУРА

1. *Werner-Allen*, *Lorincz.* Deploying a Wireless Sensor Network on an Active Volcano // IEEE Internet Computing. — 2006. — Vol. 10, N 2.
2. *Liyang Yu*, *Neng Wang, Xiaoqiao Meng.* Real-time forest fire detection with wireless sensor networks // Wireless Communications, Networking and Mobile Computing. — 2005. — Vol. 2. — P. 1214—1217.
3. *Martinez K.*, *Padhy P.*, *Riddoch A.*, et al. Glacial Environment Monitoring using Sensor Networks. URL: http://www.sics.se/realwsn05/papers/martinez05glacial.pdf (дата обращения 13.12.2010).
4. *Akkaya K.*, *Younis F.* A Survey on Routing Protocols for Wireless Sensor Networks // Ad Hoc Networks. — 2005. — Vol. 3, N 3. — P. 325—349.
5. *Трифонов С.В.*, *Истомин Т.Е.*, *Чечендаев А.В.* Алгоритмы оптимизации работы беспроводной сенсорной сети на базе протокола ZigBee // Тр. V Всеросс. межвуз. конф. молодых ученых. — СПб., 2008.
6. *IEEE TG 15.4.* Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE standard for Information Technology, IEEE-SA Standards Board, 2003.
7. *ZigBee* Specification. — ZigBee Alliance, 2006.
8. *Alon N.* The Number of Spanning Trees in Regular Graphs // Random Structures and Algorithms. — 1990. — N 1.
9. *Garg N.*, *Konemann J.* Faster and simpler algorithms for multicommodity flows and other fractional packing problems // Proc. 39th Annual Symposium on the Foundations of Computer Science, 1998. — P. 300—309.
10. *LPSOLVE* project. URL: http://sourceforge.net/projects/lpsolve/ (дата обращения 13.12.2010).

Статья представлена к публикации членом редколлегии А.С. Манделем.

Наградов Евгений Александрович — аспирант,
Московский государственный университет
им. М.В. Ломоносова,
✉ nea@lvk.cs.msu.su.