ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Ноябрь—декабрь 2011. Том 18, № 6. С. 17–32

УДК 519.8

О МАКСИМИЗАЦИИ ВРЕМЕНИ ФУНКЦИОНИРОВАНИЯ СЕНСОРНЫХ СЕТЕЙ ПРИ РЕСУРСНЫХ ОГРАНИЧЕНИЯХ [*)]

А. И. Ерзин, Р. В. Плотников

Аннотация. Рассматривается задача максимизации времени жизни сенсорной сети в условиях ограниченности ресурсов сенсоров в виде задачи целочисленного линейного программирования, в которой при заданном множестве покрытий требуется определить время функционирования каждого покрытия. При этом ресурс сенсора задаётся количеством временных раундов, в течение которых он может находиться в активном состоянии. Доказана NP-трудность задачи в сильном смысле. Предложены способы её упрощения. Показано, что для любого $\varepsilon > 0$ задача в общем случае не аппроксимируема полиномиальными алгоритмами с точностью $O(m^{1-\varepsilon})$, где m — число покрытий. Найдены частные случаи, когда задача полиномиально разрешима. Предложено несколько эвристических алгоритмов построения приближённого решения задачи и проведён апостериорный анализ.

Ключевые слова: сенсорная сеть, максимизация времени жизни, расход энергии, целочисленное линейное программирование.

Введение

Сенсор — автономное интеллектуальное устройство, которое предназначено для сбора, обработки, получения и передачи данных. Сенсор может находиться либо в активном состоянии, выполняя свои функции и расходуя энергию, либо в состоянии сна, когда расходом энергии можно пренебречь. В беспроводной сенсорной сети (СС) каждый сенсор обладает ограниченным невозобновляемым запасом энергии — ресурсом, который часто измеряют в количестве временных раундов, в течение которых сенсор может находиться в активном состоянии. Так как число сенсоров в СС существенно превышает минимальное количество, необходимое для сбора и обработки данных, функции сенсорной сети может

[*)] Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 10–07–92650–IND–a) и ФЦП «Научные и научно-педагогические кадры инновационной России» (грант № 14.740.11.0362).

выполнять подмножество её элементов. Основной функцией СС является мониторинг, и объект (точка области) считается покрытым, если он находится в зоне мониторинга хотя бы одного сенсора. В связи с этим в литературе подмножество сенсоров, выполняющее функции сети, называют *покрытием.* Хотя один и тот же сенсор может входить в разные покрытия, общее время функционирования сенсора ограничено его ресурсом. Основной задачей СС является оптимизация энергопотребления, что влечёт увеличение времени функционирования (жизни) СС. Таким образом, одним из способов максимизации времени жизни СС является определение времени функционирования каждого покрытия с учётом максимального времени функционирования каждого сенсора.

Итак, *покрытием* назовём подмножество сенсоров, обеспечивающее функционирование СС. В зависимости от приложения цели СС могут различаться. В некоторых случаях требуется покрыть заданную область, либо её часть, в других — множество объектов. Часто на покрытия накладываются дополнительные ограничения (связность [15], наличие определённой структуры [1, 18] и др.).

В [5] задача максимизации времени жизни СС в случае, когда множество покрытий не задано, а переменные, соответствующие времени функционирования покрытий, непрерывны, сформулирована в виде частного случая задачи линейного программирования (packing linear program) и предлагается метод её решения, основанный на алгоритме Гарга — Кёнеманна (Garg–Konemann) [10]. В [9] рассматривается более общая задача с регулируемыми радиусами мониторинга в аналогичной постановке. Для её приближённого решения авторам удалось обобщить метод, предложенный в [5]. В [16] рассматривается задача максимизации времени жизни СС в случае, когда ресурсы всех вершин одинаковые. Для приближённого решения задачи предложен алгоритм, который строит множество покрытий с минимальным количеством многократно покрываемых объектов, используя метод ветвей и границ, и работает, пока сенсоры, ресурс которых не израсходован, покрывают все объекты. При этом время функционирования одного покрытия задано входным параметром w. В ходе численного эксперимента предложенный алгоритм сравнивался с алгоритмом Greedy–MSC [8]. В [7] для каждого сенсора задано множество объектов, которые он покрывает, и требуется построить множество непересекающихся покрытий, суммарное время жизни которых максимально.

В нашей статье рассматривается задача, аналогичная по постановке задаче, сформулированной в [5], в условиях заданного избыточного множества покрытий и целочисленных переменных, соответствующих числу временных раундов, в течение которых покрытие функционирует.

В разд. 1 сформулирована рассматриваемая задача максимизации времени жизни СС в виде задачи целочисленного линейного программирования (ЦЛП). В разд. 2 доказана NP-трудность этой задачи. В разд. 3 предложены различные способы упрощения задачи. В разд. 4 задача исследуется на аппроксимируемость: показано отсутствие аппроксимационной схемы в общем случае, а также найдены условия, при которых задача принадлежит классу APX. В разд. 5 рассмотрены частные случаи, когда задача полиномиально разрешима, и предлагаются соответствующие алгоритмы. В разд. 6 предложено два эвристических алгоритма для приближённого решения задачи в общем случае и проведён численный эксперимент, показавший высокую эффективность предложенных алгоритмов.

1. Постановка задачи

Пусть J — множество сенсоров, $|J| = n$, и каждый сенсор $j \in J$ обладает ограниченным ресурсом $r_j \in \mathbb{Z}^+$. Предположим, что задано множество различных покрытий $C = \{C_1, \ldots, C_m\}$, $C_i \subseteq J$. Введём параметры $b_{jk} = 1$, если сенсор $j \in C_k$, и $b_{jk} = 0$ в противном случае, а также вектор переменных $y = (y_1, \ldots, y_m)$, где $y_k \in \mathbb{Z}^+$ — время функционирования (жизни) покрытия C_k (т.е. число временных раундов, в течении которых элементы покрытия C_k активны). Тогда задача максимизации времени жизни СС запишется следующим образом:

$$\sum_{k=1}^{m} y_k \to \max_{y_k \in Z_0^+}, \tag{1}$$

$$\sum_{k=1}^{m} b_{jk} y_k \leqslant r_j, \quad j \in J. \tag{2}$$

Для каждого покрытия $C_k \in C$ определим его ресурс r^k, равный минимальному начальному ресурсу входящих в него сенсоров. Очевидно, что величина r^k ограничивает сверху время жизни покрытия C_k. Минимальный и максимальный начальные ресурсы сенсоров множества J обозначим через $r_{\min}$ и $r_{\max}$ соответственно. Назовём сенсор *дефицитным*, если его ресурс меньше суммы ресурсов содержащих его покрытий.

2. Доказательство NP-трудности

Докажем, что задача (1)–(2) NP-трудна в сильном смысле. Рассмотрим частный случай задачи (1)–(2) (когда ресурс каждого сенсора единичный):

$$\sum_{k=1}^{m} y_k \to \max_{y_k \in \{0,1\}}, \tag{3}$$

$$\sum_{k=1}^{m} b_{jk} y_k \leqslant 1, \quad j \in J. \tag{4}$$

Перед формулировкой леммы 1 напомним известную задачу Максимальное независимое множество (МНМ): *для заданного графа найти независимое подмножество вершин максимальной мощности.*

Лемма 1. *Задачи* (3)–(4) *и* МНМ *полиномиально эквивалентны.*

Доказательство. Задача (3)–(4) состоит в поиске максимального по мощности множества непересекающихся покрытий. Построим следующий граф пересечений. Каждому покрытию C_k поставим в соответствие вершину этого графа. Две вершины смежны тогда и только тогда, когда соответствующие покрытия имеют общие сенсоры. Тогда задача (3)–(4) сводится к МНМ процедурой построения графа пересечений покрытий. Очевидно, такой граф может быть построен за полиномиальное время.

Для доказательства обратной полиномиальной сводимости рассмотрим произвольную индивидуальную МНМ в некотором графе $G = (V, E)$. Требуется найти независимое подмножество вершин максимальной мощности. Считаем, что изолированных вершин в графе нет, иначе их можно исключить, так как они заведомо войдут в максимальное независимое множество. Покажем, что данная задача является альтернативной записью некоторой индивидуальной задачи (3)–(4). Действительно, пусть вершина $k \in V$ соответствует покрытию C_k, ребро $e_j \in E$ соответствует сенсору j, принадлежащему покрытию C_k тогда и только тогда, когда соотвествующие вершина и ребро графа G инцидентны. Требование независимости подмножества вершин графа G эквивалентно условию (4) для соответствующих сенсоров и покрытий, где $m = |V|$, а мощность независимого подмножества вершин равна суммарному времени жизни соответствующих покрытий. Следовательно, сведение задачи МНМ к задаче (3)–(4) заключается в её эквивалентной записи в виде задачи ЦЛП на матрице инцидентности рассматриваемого графа. Лемма 1 доказана.

Следствие 1. *Задача* (1)–(2) NP-*трудна в сильном смысле.*

Доказательство. Известно, что задача МНМ NP-трудна в сильном смысле [2]. В доказательстве леммы 1 показано, что МНМ полиномиально сводится к задаче (1)–(2). Следовательно, задача (1)–(2) также NP-трудна в сильном смысле. Следствие 1 доказано.

3. Упрощение сенсорных сетей

Иногда СС содержит сенсоры, которые можно исключить, а также иногда удаётся сократить множество покрытий, не уменьшая времени жизни СС. Рассмотрим следующие случаи.

Случай 1. $C_j \subseteq C_i$. Очевидно, что в этом случае покрытие C_i можно удалить, так как при его активности расходуются ресурсы как вершин покрытия C_j, так и других сенсоров, входящих в C_i.

Будем говорить, что вектор $a = (a_1, \ldots, a_n)$ *доминирует* вектор $b = (b_1, \ldots, b_n)$, если $a_i \leqslant b_i$ для всех $i \in \{1, \ldots, n\}$. Заметим, что описанное выше исключение покрытий равносильно удалению из матрицы ограничений задачи (1)–(2) доминируемых столбцов.

Случай 2. Два сенсора входят в одни и те же покрытия. В этом случае их ресурсы расходуются одновременно, поэтому сенсор с бóльшим ресурсом может быть (виртуально) исключён из J. Следовательно, если k сенсоров принадлежат одним и тем же покрытиям, то из них можно оставить один сенсор с минимальным ресурсом.

Случай 3. Недефицитный сенсор можно исключить из J. Если сенсор принадлежит более чем одному покрытию и величина его ресурса не меньше суммы ресурсов этих покрытий, то такой сенсор можно исключить. Возможно, после такого исключения некоторые покрытия перестанут пересекаться.

После применения описанных выше правил останутся только дефицитные сенсоры, в матрице ограничений $B = ||b_{ij}||$ все строки будут различны и ни один столбец не будет доминировать какой-либо другой столбец.

4. Аппроксимируемость задачи

Пусть заданы фиксированные параметры a, b и δ: $0 < a \leqslant b$, $\delta \geqslant 3$. Рассмотрим задачу $P_{a,b,\delta}$:

$$\sum_{k=1}^{m} y_k \to \max_{y_k \in Z_0^+},$$

$$\sum_{k=1}^{m} b_{jk} y_k \leqslant r_j, \quad j \in \{1, \ldots, n\},$$

$$a \leqslant r_j \leqslant b, \quad j \in \{1, \ldots, n\},$$

$$\sum_{l=1,\, l \neq k}^{m} \operatorname{sgn}\left(\sum_{j=1}^{n} b_{jl} b_{jk} \right) \leqslant \delta, \quad k \in \{1, \ldots, m\}.$$

В этой постановке ресурс каждого сенсора принадлежит фиксированному отрезку $[a,\ b]$ и каждое покрытие пересекается не более чем с δ другими покрытиями.

Лемма 2. *Задача $P_{a,b,\delta}$ принадлежит классу* APX.

Доказательство. Рассмотрим задачу МНМ для графа, степень которого равна δ, и обозначим её через МНМ-δ. Известно [4], что эта задача принадлежит классу APX: для любого фиксированного значения $\varepsilon > 0$ найдётся такой полиномиальный алгоритм A_ε, что для любой индивидуальной задачи МНМ-δ выполняется неравенство

$$\frac{F_\varepsilon}{F^*} \geqslant \frac{5}{\delta + 3 + \varepsilon}, \tag{5}$$

где F_ε — мощность независимого множества, построенного алгоритмом A_ε, F^* — оптимальное значение функционала. Для произвольной индивидуальной задачи $I \in P_{a,b,\delta}$ рассмотрим граф пересечений покрытий G. Степень этого графа не превосходит δ. Пусть $\varepsilon > 0$. Применим алгоритм A_ε к МНМ-δ в графе G. Пусть $\widetilde{F}_\varepsilon$ — мощность независимого множества вершин графа, построенного алгоритмом A_ε, $\widetilde{F}^*$ — мощность максимального независимого множества графа G. Согласно (5) выполняется неравенство

$$\frac{\widetilde{F}_\varepsilon}{\widetilde{F}^*} \geqslant \frac{5}{\delta + 3 + \varepsilon}.$$

Алгоритм A_ε выбирает непересекающиеся покрытия, а значит, строит допустимое решение задачи I. Пусть W_ε — время жизни СС, определяемое построенным решением, а W^* — максимальное время жизни СС. Время жизни каждого выбранного покрытия равно ресурсу этого покрытия, поэтому $W_\varepsilon \geqslant r_{\min} \widetilde{F}_\varepsilon$. Пусть k — хроматическое число графа G. Тогда $\widetilde{F}^* \geqslant \frac{m}{k}$. Известно [3], что $k \leqslant \delta + 1$. Поэтому $\widetilde{F}^* \geqslant \frac{m}{\delta+1}$. Так как $W^* \leqslant r_{\max} m$, то $W^* \leqslant (\delta + 1) r_{\max} \widetilde{F}^*$ и

$$\frac{W_\varepsilon}{W^*} \geqslant \frac{r_{\min} \widetilde{F}_\varepsilon}{(\delta + 1) r_{\max} \widetilde{F}^*} \geqslant \frac{5a}{(\delta + 1)(\delta + 3 + \varepsilon) b}.$$

Поскольку значения a, b и δ фиксированы, в правой части последнего неравенства находится константа, и, следовательно, $P_{a,b,\delta} \in$ APX. Лемма 2 доказана.

Следствие 2. *Любая общая задача, состоящая из конечного числа индивидуальных задач проблемы* (1)–(2), *принадлежит классу* APX.

Лемма 3. *Если $P \neq NP$, то для любого $\varepsilon > 0$ не существует полиномиального алгоритма A, приближённо решающего задачу* (1)–(2), *для которого верна оценка точности*

$$\frac{F_A}{F^*} = O(m^{1-\varepsilon}),$$

где F_A — время жизни СС, *полученное алгоритмом A, F^* — максимальное время жизни* СС.

Доказательство. В [13] доказано, что при условии P≠NP задача о максимальной клике (МК) не может быть полиномиально аппроксимирована с точностью $O(m^{1-\varepsilon})$ при произвольном $\varepsilon > 0$. Так как задачи МК и МНМ эквивалентны, а задача МНМ полиномиально эквивалентна подклассу задачи (1)–(2), задача (1)–(2) не аппроксимируема с коэффициентом $O(m^{1-\varepsilon})$ для любого $\varepsilon > 0$. Лемма 3 доказана.

5. Полиномиально разрешимые случаи

В этом разделе рассмотрим частные случаи, когда задача (1)–(2) полиномиально разрешима.

5.1. Абсолютная унимодулярность матрицы ограничений. Проблема (1)–(2) является задачей ЦЛП, поэтому в случае, когда матрица ограничений B абсолютно унимодулярна (АУ), задача может быть решена за полиномиальное время [12]. Найдём достаточные условия абсолютной унимодулярности матрицы B. Напомним следующие критерии АУ.

(i) Целочисленная матрица A размера $n \times m$ абсолютно унимодулярна тогда и только тогда, когда для всякого подмножества строк $R \subset \{1, \ldots, n\}$ найдётся такое его разбиение на два непересекающихся множества R_1 и R_2, что выполняются неравенства [12]

$$\Big| \sum_{j \in R_1} a_{jk} - \sum_{j \in R_2} a_{jk} \Big| \leqslant 1, \quad k \in \{1, \ldots m\}. \tag{6}$$

(ii) Матрица A абсолютно унимодулярна тогда и только тогда, когда транспонированная матрица A^T абсолютно унимодулярна.

Лемма 4. *Задача* (1)–(2) *полиномиально разрешима, если множество сенсоров можно разбить на два таких подмножества, что сенсоры одного подмножества принадлежат разным покрытиям.*

Доказательство. Пусть найдётся разбиение множества сенсоров, удовлетворяющее условиям леммы. Очевидно, данное разбиение делит любое подмножество $J' \subseteq J$ на два множества J_1 и J_2, для которых выполняется условие (6). Следовательно, матрица ограничений B абсолютно унимодулярна. Лемма 4 доказана.

Очевидно, что для выполнения условий леммы 4 необходимо, чтобы в каждое покрытие входило не более двух сенсоров. Кроме того, покрытия, содержащие ровно один сенсор, могут быть исключены из множества C (см. разд. 3). В таком случае удобно рассматривать СС в виде графа $G' = (J, C)$, вершины которого соответствуют сенсорам, а рёбра — покрытиям. Из леммы 4 следует, что для полиномиальной разрешимости задачи (1)–(2) достаточно, чтобы граф G' был двудольным.

Замечание 1. Если каждое покрытие из C содержит 2 сенсора, ресурс каждого сенсора равен 1 и граф G' двудолен, то задача (1)–(2) эквивалентна задаче поиска максимального паросочетания в двудольном графе. Из известных алгоритмов, решающих задачу построения максимального паросочетания в двудольном графе, наиболее эффективными являются алгоритм Хопкрофта — Карпа [14], имеющий трудоёмкость $O(m\sqrt{n})$, и алгоритм Мучи — Санковского [17], трудоёмкость которого равна $O(n^{2.376})$.

Лемма 5. *Если каждое покрытие из C содержит 2 сенсора и граф G' является деревом, то решение задачи* (1)–(2) *строится алгоритмом* A1 *с трудоёмкостью $O(n^2)$.*

Алгоритм A1
$(y_1, \ldots, y_m) := (0, \ldots, 0)$; $S := \{C_1, \ldots, C_m\}$;
пока $S \neq \varnothing$ **выполнить**
  поиск всех висячих вершин G'; к каждой найденной висячей вершине i применение процедуры Отсечение висячей вершины(i);

Отсечение висячей вершины(i)
Найдём вершину j — родителя вершины i в графе G'.
Найдём такой индекс k, что покрытие C_k соответствует ребру (i, j) в графе G'.
Положим $y_k := \min(r_i, r_j)$; $r_j := r_j - y_k$; $S := S \setminus C_k$.

Доказательство. Докажем, что построенное алгоритмом A1 решение оптимально. Для этого достаточно проверить, что процедура отсечение висячей вершины не ухудшает решение. Для простоты изложения и без ограничения общности будем считать, что 1 — некоторая висячая вершина в графе G', вершина 2 — её родитель, C_1 — покрытие, соответствующее ребру (1, 2), и $\{C_1, C_2, \ldots, C_s\}$ — покрытия, содержащие вершину 2. Пусть $(y_1^*, \ldots, y_m^*)$ — оптимальное решение задачи(1)–(2). Пусть $y_1' = \min(r_1, r_2)$. Допустим, что процедура отсечение висячей вершины, применённая к вершине 1, уменьшает значение целевой функции. Тогда для любых допустимых $y_1, \ldots, y_s$ выполняется неравенство

$$y_1' + \sum_{y=2}^{s} y_k < \sum_{y=1}^{s} y_k^*.$$

Последовательно найдём значения величин $y_2', \ldots, y_s'$, полагая

$$y_l' = \min\left(y_l^*, r_2 - \sum_{k=1}^{l-1} y_k'\right).$$

Так как решение $(y_1^*, y_2^*, \ldots, y_m^*)$ оптимально, вектор $(y_1', \ldots, y_s', y_{s+1}^*, \ldots, y_m^*)$ допустимый. Имеем

$$\sum_{k=2}^{s} y_k' = \min\left(\sum_{k=2}^{s} y_k^*, r_2 - y_1'\right).$$

Следовательно,

$$\sum_{k=1}^{s} y_k' = \min\left(y_1' + \sum_{k=2}^{s} y_k^*, r_2\right).$$

Так как $y_1^* \leqslant \min(r_1, r_2) = y_1'$ и $r_2 \geqslant \sum\limits_{k=1}^{s} y_k^*$, то $\sum\limits_{k=1}^{s} y_k' \geqslant \sum\limits_{k=1}^{s} y_k^*$, и получаем противоречие с предположением о том, что процедура отсечение висячей вершины уменьшает значение функционала. Значит, из оптимальности решения $(y_1^*, y_2^*, \ldots, y_m^*)$ следует оптимальность решения $(y_1', \ldots, y_s', y_{s+1}^*, \ldots, y_m^*)$.

Лемма 6. *Если граф пересечений покрытий является двудольным, то задача* (1)–(2) *полиномиальна разрешима.*

Доказательство. Если граф пересечений покрытий двудолен, то множество покрытий C разбивается на два непересекающихся подмножества таким образом, что никакие два покрытия из одного подмножества не содержат общих сенсоров. Следовательно, условие (6) выполняется для любого подножества строк матрицы B^T. Значит, матрица B абсолютно унимодулярна. Лемма 6 доказана.

Доказательство следующей леммы носит технический характер, но достаточно длинное, поэтому мы его опустим.

Лемма 7. *Если граф пересечений покрытий G является деревом и ресурсы всех вершин* СС *совпадают, то решение задачи* (1)–(2) *строится с трудоёмкостью* $O(m^2)$.

5.2. Частные случаи. Рассмотрим следующие частные случаи, когда задача (1)–(2) может быть решена с полиномиальной трудоёмкостью.

Случай 1. Пусть каждое покрытие пересекается с другими покрытиями не более чем по одному сенсору. Тогда множество покрытий разбивается на непересекающиеся подмножества, определяемые общей вершиной. Это разбиение делит исходную задачу на независимые подзадачи, поэтому решение всей задачи состоит в последовательном решении подзадач. Заметим, что в каждой подзадаче время жизни СС определяет дефицитный сенсор, и оптимальное решение состоит в последовательном использовании произвольного покрытия с положительным ресурсом.

Случай 2. Каждое покрытие пересекается не более чем с одним другим покрытием. Тогда множество покрытий разбивается на подмножества, каждое из которых состоит либо из двух, либо из одного покрытия. Подзадачи с одним покрытием тривиальны, поэтому имеет смысл рассматривать лишь подзадачи с двумя пересекающимися покрытиями. Оптимальное решение всей исходной задачи может быть найдено с трудоёмкостью $O(n+m)$ последовательным решением подзадач.

Случай 3. Частные случаи задачи (3)–(4). Если ресурсы всех вершин равны единице, то, как доказано в разд. 2, задача эквивалентна МНМ. Известно, что МНМ полиномиально разрешима, в частности, в следующих случаях [11]:

(i) граф G является рёберным;
(ii) граф G является хордальным;
(iii) степень графа G равна 2.

Любая индивидуальная задача (3)–(4) может быть решена за полиномиальное время, если соответствующий граф пересечений покрытий удовлетворяет приведённым выше свойствам.

6. Приближённые алгоритмы

В этом разделе предложены два эвристических алгоритма для приближённого решения задачи (1)–(2), а также проведён апостериорный анализ этих алгоритмов.

6.1. Описание алгоритмов. На предварительном этапе каждого алгоритма осуществляется упрощение задачи по правилам, описанным в разд. 3. Затем сенсоры упорядочиваются по неубыванию дефицита $d_j = \sum\limits_{k=1}^{m} b_{jk} - r_j$. После чего для каждого сенсора j определяется список V_j, состоящий из номеров покрытий, содержащих этот сенсор, и элементы списка V_j упорядочиваются по невозрастанию ресурсов покрытий. В ходе работы алгоритма H1 каждый сенсор последовательно тратит по единице ресурса в каждом из покрытий, его содержащих, после чего ресурсы других покрытий пересчитываются. В ходе работы алгоритма H2 ресурс каждого сенсора используется содержащими его покрытиями пропорционально их ресурсам. При этом если деление ресурса сенсора нацело между покрытиями невозможно, то остаток от деления распределяется между первыми выбранными покрытиями (по единице ресурса на покрытие). Алгоритм H1 имеет псевдополиномиальную трудоёмкость $O\big(mn \sum\limits_{k=1}^{m} y_k\big) \leqslant O(r_{\max} n^2 m)$. Трудоёмкость алгоритма H2 равна $O(n^2m^2)$.

Алгоритм H1

$(y_1, \ldots, y_m) := (0, \ldots, 0)$; упростим задачу, используя правила разд. 3; для каждого сенсора $j \in J$ вычислим дефицит $d_j := \sum\limits_{k=1}^{m} b_{jk} - r_j$; упорядочим сенсоры по неубыванию дефицитов;
для любого $j \in J$ **выполнить**
  $V_j := \{C_k \in C : b_{jk} = 1\}$; упорядочим элементы V_j по невозрастанию ресурсов;
$R := \sum\limits_{k=1}^{m} r^k$;
пока $R > 0$ **выполнить**
  для любого $j \in J$ **выполнить**
    для любого $C_k \in V_j$ **выполнить**
      если $r^k > 0$ **то**
        $y_k := y_k + 1$; $r^k := r^k - 1$;
        для любого $j_k \in C_k$ **выполнить**
          $r_{j_k} := r_{j_k} - 1$;

для любого $l \in \{1, \ldots, m\}$: $C_l \cap C_k \neq \varnothing$ **выполнить**
пересчёт r^l;

Алгоритм H2
$(y_1, \ldots, y_m) := (0, \ldots, 0)$; упростим задачу, используя правила разд. 3;
для каждого сенсора $j \in J$ вычислим дефицит $d_j := \sum_{k=1}^{m} b_{jk} - r_j$; упорядочим сенсоры по неубыванию дефицитов;
для любого $j \in J$ **выполнить**
$V_j := \{C_k \in C : b_{jk} = 1\}$; упорядочение элементов V_j по невозрастанию ресурсов;
для любого $j \in J$ **выполнить**
найдём S — сумму ресурсов покрытий из V_j;
для любого $k \in \{1, \ldots, m\}$ **выполнить**
$p_k := \left\lfloor r_j \left(r^k / S\right)\right\rfloor$;
найдём остаток ресурса сенсора j: $q := r_j - \sum_{k: C_k \in V_j} p_k$;
для любого $k \in V_j$ **выполнить**
если $r^k > 0$ **то**
если $r^k > p_k$ **то**
$d := \operatorname{sgn}(q)$; $x := p_k + d$; $q := q - d$;
иначе
$x := r^k$; $q := q + (p_k - r^k)$;
$y_k := y_k + x$;
$r^k := r^k - x$;
для любого $j_k \in C_k$ **выполнить**
$r_{j_k} := r_{j_k} - x$;
для любого $l \in \{1, \ldots, m\}$: $C_l \cap C_k \neq \varnothing$ **выполнить**
пересчёт r^l;

6.2. Численный эксперимент. Алгоритмы H1 и H2 реализованы на языке программирования C++. Численные эксперименты проведены на случайно сгенерированных тестовых примерах. В качестве области мониторинга использовался квадрат со стороной 100. Каждый сенсор случайно размещался внутри рассматриваемой области. Для каждого покрытия генерировалась регулярная треугольная решётка, располагающаяся в области случайным образом. Сторона треугольника решётки d задавалась на входе. Каждое покрытие соответствовало одной решётке и состояло из сенсоров, ближайших к узлам решётки. Мы полагали, что узел решётки покрыт, если расстояние от него до ближайшего сенсора не превосходило d, а вся область покрыта, если покрыты все узлы решётки.

Очевидно, что чем меньше d, тем больше сенсоров следует разместить. Мы использовали 6 вариантов тестов, в каждом из которых величины d и n были сбалансированы. Первый вариант: $d = 35$ и $n = 25$, второй вариант: $d = 25$ и $n = 50$, третий вариант: $d = 10$ и $n = 100$, четвёртый вариант: $d = 15$ и $n = 200$, пятый вариант: $d = 10$ и $n = 500$ и шестой вариант: $d = 5$ и $n = 1000$. Для каждого примера найдена покрытая часть области (CP) — доля покрытых узлов решётки. Для оценки зависимости между покрытиями для каждого сенсора найдено число содержащих его покрытий (NC). Эксперименты проведены с одинаковыми для всех сенсоров ресурсами (20), со случайно заданным между 1 и 5 ресурсом для каждого сенсора и со случайно заданным между 10 и 50 ресурсом для каждого сенсора. Оптимальное решение строилось программой IBM CPLEX, предоставленной в рамках Соглашения для академической инициативы IBM. Пример одной размерности запускался 100 раз. Точность каждого алгоритма оценивалась средней величиной отношения F_A/F^*, где F_A — значение функционала на построенном алгоритмом A решении, а F^* — максимальное значение функционала. Средние значения величин NC и CP также посчитаны для каждого случая.

Мы сравнили полученные результаты с двумя простыми методами. Первый — жадный алгоритм (ЖА), в котором на каждом шаге выбирается покрытие с максимальным ресурсом, после чего ресурсы покрытий пересчитываются. Второй — симплекс-метод с округлением (СО), т. е. взятие целых частей оптимального решения непрерывной задачи (1)–(2). Результаты проведённых экспериментов отображены в табл. 1.

В большинстве случаев оба предложенных алгоритма работают лучше жадного, но алгоритм H1 часто работает лучше алгоритма H2. Если ресурсы сенсоров достаточно велики, то хорошее решение может быть получено с помощью округления решения, построенного симплекс-методом. Однако этот метод работает плохо при малых значениях ресурса.

Заключение

Рассмотрена задача максимизации времени жизни сенсорной сети при заданном избыточном множестве покрытий в условиях ограниченности ресурсов сенсоров. Она сформулирована в виде задачи целочисленного линейного программирования. Доказана NP-трудность этой задачи и отсутствие для неё аппроксимационной схемы. Предложены способы упрощения задачи, найдены частные случаи, когда задача полиномиально разрешима, и условия, при которых она принадлежит классу APX. Для приближённого решения задачи в общем случае предложено два эвристических алгоритма. Для оценки качества алгоритмов проведён численный эксперимент, в ходе которого результаты их работы сравнивались с результатами, полученными жадным алгоритмом и округлением решений непрерывной задачи. Для получения оптимального решения задачи использовался пакет программ IBM CPLEX, предоставленный авторам в соответствии с Соглашением для академической инициативы IBM.

Таблица 1

Результаты эксперимента

n	m	d	$r_{\min}$	$r_{\max}$	NC	CP (%)	ratio(CO)	ratio(ЖА)	ratio(H1)	ratio(H2)
25	50	35	1	5	19.5	99.2	0.834	0.797	0.804	0.792
25	10	35	20	20	4.45	99.3	0.968	0.833	0.894	0.823
25	50	35	20	20	19.9	99.2	0.956	0.622	0.752	0.654
25	50	35	10	50	19.3	99.3	0.981	0.824	0.792	0.768
50	10	25	1	5	4.08	99.3	0.832	0.832	0.857	0.857
50	50	25	1	5	18.8	99.5	0.652	0.762	0.751	0.751
50	10	25	20	20	4.07	99.4	0.983	0.949	0.97	0.949
50	50	25	20	20	20.9	99.2	0.905	0.633	0.731	0.655
50	10	25	10	50	4.01	99.7	0.986	0.888	0.87	0.846
50	50	25	10	50	18.9	99.6	0.942	0.782	0.782	0.734
100	10	20	1	5	3.27	100	0.717	0.852	0.827	0.827
100	50	20	1	5	15.1	99.9	0.388	0.753	0.716	0.716
100	10	20	20	20	3.43	99.9	0.948	0.767	0.888	0.767
100	50	20	20	20	15.3	99.9	0.89	0.572	0.75	0.572
100	10	20	10	50	3.23	99.8	0.981	0.838	0.87	0.819
100	50	20	10	50	14.7	99.8	0.934	0.763	0.774	0.714
200	10	15	1	5	2.94	100	0.465	0.827	0.818	0.818
200	50	15	1	5	13.5	99.9	0.182	0.677	0.741	0.741
200	10	15	20	20	3.16	99.9	0.918	0.728	0.852	0.728
200	50	15	20	20	13.5	100	0.868	0.574	0.782	0.574
200	10	15	10	50	2.93	100	0.953	0.77	0.861	0.749
500	10	10	1	5	2.64	100	0.32	0.87	0.9	0.9
500	50	10	1	5	11.6	100	0.0333	0.687	0.673	0.673
500	10	10	20	20	2.87	99.9	0.908	0.74	0.848	0.74
500	50	10	20	20	12.6	100	0.826	0.597	0.815	0.597
500	10	10	10	50	2.62	100	0.933	0.681	0.866	0.671
500	50	10	10	50	12.5	100	0.827	0.632	0.75	0.608
1000	10	5	20	20	4.81	99	0.988	1	1	1
1000	50	5	20	20	23	99	0.99	1	1	1
1000	10	5	10	50	5.2	98.9	0.957	0.911	0.978	0.911
1000	50	5	10	50	24.1	98.8	0.872	0.838	0.919	0.835

В дальнейшем планируется разработать эффективные методы построения покрытий, а также учесть зависимость расхода энергии сенсора от площади покрываемой им области и дальности передачи.

ЛИТЕРАТУРА

1. **Астраков С. Н., Ерзин А. И., Залюбовский В. В.** Сенсорные сети и покрытие плоскости кругами // Дискрет. анализ и исслед. операций. — 2009. — Т. 16, № 3. — С. 3–19.
2. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. — М.: Мир, 1982. — 416 с.
3. **Харари Ф.** Теория графов. — М.: Мир, 1973. — 296 с.
4. **Berman P., Fujito T.** Approximating independent sets in degree 3 graphs // Proc. 4th Workshop Algorithms Data Structures. — Berlin: Springer-Verl., 1995. — P. 449–460. (Lect. Notes Comput. Sci.; Vol. 955.)
5. **Berman P., Calinescu G., Shan C., Zelikovsky A.** Power efficient monitoring management in sensor networks // Proc. IEEE Wireless Communications Networking Conf. (Atlanta, USA, March 21–25, 2004). — Los Alamitos, CA: IEEE Comp. Soc., 2006. — P. 2329–2334.
6. **Berman P. Karpinski M.** On some tighter inapproximability results // Tech. Rep. TR98–065, ECCC, 1998.
7. **Cardei M., Ding-Zhu D.** Improving wireless sensor network lifetime through power aware organization // Springer Sci., Business Media, Wireless Networks 11, Netherlands, 2005 — P. 333–340.
8. **Cardei M., Thai M. T., Li Y., Wu W.** Energy-efficient target coverage in wireless sensor networks // IEEE Infocom. — 2005. — Vol. 3. — P. 1976–1984.
9. **Dhawan A., Vu C. T., Zelikovsky A., Li Y., Prasad S. K.** Maximum lifetime of sensor networks with adjustable sensing range // Proc. 7th ACIS Int. Conf. Software Engineering (Las Vegas, Nevada, USA, June 19–20, 2006). — Los Alamitos: IEEE Comp. Soc., 2006. — P. 285–289.
10. **Garg N., Konemann J.** Faster and simpler algorithms for multicommodity flow and other fractional packing problems // Proc. FOCS (Palo Alto, CA, USA, November 8–10, 1998). —Washington: IEEE Comp. Soc., 1998. — P. 300–309.
11. **Gavril F.** Algorithms on circular-arc graphs // Networks. — 1974. — Vol. 4. — P. 357–369.
12. **Ghouila-Houri A.** Caractérisation des matrices totalement unimodulaires // CR hebdomadaires des séances de l'Académie des sciences, Paris, 1962. — P. 1192–1194.
13. **Hastad G.** Clique is hard to approximate within $n^{1-\varepsilon}$ // Tech. Rep. TR97–038, ECCC, 1997.
14. **Hopcroft J., Karp R.** An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs // SIAM J. Comput. — 1973. — Vol. 2. — P. 225—231.
15. **Inanc M., Magdon-Ismail M., Yener B.** Power optimal connectivity and coverage in wireless sensor networks // Department of Computer Science,

Rensselaer Polytechnic Institute. — Troy, NY: Rensselaer Polytech. Inst. Publ., 2003.

16. **Kim Y., Lee H., Han Y. Jeonge Y.** A branch and bound algorithm for extending the lifetime of wireless sensor networks // Proc. Vehicular Tech. Conf. (Anchorage, AK, September 20–23, 2009). — Washington: IEEE Comp. Soc., 2009. — P. 1–5.
17. **Mucha M., Sankowski P.** Maximum matchings via Gaussian elimination // Proc. 45th IEEE Symp. Foundations Comput. Sci. 2004 (Washington, DC, USA, October 17–19, 2004) . — Washington: IEEE Comp. Soc., 2004. — P. 248—255.
18. **Segal M.** Improving lifetime of wireless sensor networks // Netw. Protocols Algorithms. — 2009. — Vol. 1, N 2. — P 48–60.

Ерзин Адиль Ильясович,
e-mail: adilerzin@math.nsc.ru
Плотников Роман Викторович,
e-mail: nomad87@ngs.ru

Статья поступила
12 апреля 2011 г.

Переработанный вариант —
16 июня 2011 г.