УДК 519.85

А.И. КОСОЛАП[*], А.С. ПЕРЕТЯТЬКО[*]

ПОЛУОПРЕДЕЛЕННАЯ ОПТИМИЗАЦИЯ В ЗАДАЧЕ РАСПОЛОЖЕНИЯ ДАТЧИКОВ В СЕТИ

[*]Украинский государственный химико-технологический университет, Днепропетровск, Украина

Анотація. *Розглядається задача розташування датчиків у мережі з мінімізацією норми відхилень від заданих відстаней. Для її розв'язку використовуються напіввизначена релаксація та новий напіввизначений симплекс-метод. Метод реалізований програмно, проведені численні експерименти, які свідчать, що напіввизначена релаксація для цього класу задач є ефективною.*
Ключові слова: *задача розташування датчиків у мережі, напіввизначена релаксація, напіввизначений симплекс-метод, напіввизначена оптимізація.*

Аннотация. *Рассматривается задача расположения датчиков в сети с минимизацией нормы отклонений от заданных расстояний. Для ее решения используются полуопределенная релаксация и новый полуопределенный симплекс-метод. Метод реализован програмно, проведены многочисленные эксперименты, которые показывают, что полуопределенная релаксация для этого класса задач является эффективной.*
Ключевые слова: *задача расположения датчиков в сети, полуопределенная релаксация, полуопределенный симплекс-метод, полуопределенная оптимизация.*

Abstract. *Sensor network localization problem with minimization of deviation norm from required distances is considered. For its solving semidefinite relaxation and new semidefinite simplex-method are used. The method was implemented and numerical experiments were performed. They show that semidefinite relaxation is efficient for this class of problems.*
Keywords: *sensor network localization problem, semidefinite relaxation, semidefinite simplex-method, semidefinite optimization.*

1. Вступление

Полуопределенная оптимизация (SDP) является новой отраслью оптимизации, начиная с 90-х годов XX века. Многие NP-сложные задачи общей квадратичной оптимизации, комбинаторной оптимизации, положительной оптимизации, теории графов, теории управления, вычислительной геометрии, некоторые технические и финансовые задачи сводятся с помощью полуопределенной релаксации к задачам SDP [1–3]. Все эти задачи могут быть сформулированы как общие квадратичные задачи, которые затем с помощью преобразования $xAx^T = Axx^T = A \bullet X$ сводятся к задачам полуопределенной оптимизации, где X – полуопределенная матрица ранга «единица» [4]. Релаксация заключается в том, что на матрицу X накладывается только условие полуопределенности. Если решением задачи SDP будет матрица ранга «единица», то такая релаксация будет точной. Во многих случаях оптимальное решение SDP-релаксации может быть преобразовано к допустимому решению исходной задачи.

Важной задачей, где используется полуопределенная релаксация, является локализация датчиков в сети. В последние годы данная проблема интенсивно исследуется [5–12]. Важность этой задачи состоит в том, что в современном информационном обществе существует проблема автоматического сбора данных. Одним из основных факторов автоматизации сбора данных являются датчики, которые обеспечивают информационные сети данными. От количества и структуры датчиков зависит эффективность обработки данных, поступающих в компьютер. Существуют системы, которые уже имеют какое-то количество датчиков с заданной структурой, и необходимо дополнить эту структуру новыми датчиками. Задача локализации датчиков в сети заключается в том, чтобы определить такую структуру датчиков, которая полностью охватывает заданный объект наблюдения. При большом количестве датчиков задача становится сложной и требует разработки новых методов полуопределенной оптимизации.

Целью данной работы является исследование полуопределенной релаксации для решения задачи локализации датчиков в сети, а также проверка целесообразности использования нового полуопределенного симплекс-метода для полученной в результате релаксации задачи полуопределенной оптимизации.

2. Постановка задачи

Рассмотрим задачу оптимального расположения датчиков в сети [11–12]. Задача локализации датчиков в сети состоит в нахождении оптимального расположения датчиков, которое удовлетворяет заданным расстояниям между ними. Пусть имеется граф $G=(V,E)$. Датчиками называют его вершины, координаты которых необходимо определить.

Пусть имеется m закрепленных вершин графа $a^1,..., a^m \in R^d$ и n вершин $x^1,..., x^n \in R^d$, чье расположение нам необходимо определить. К тому же нам известны значения евклидовых расстояний d_{ij} между a^i и x^j для некоторых i, j и $\bar{d}_{ij}$ между x^i и x^j для некоторых $i<j$. Введем обозначения: пусть $N_a=\{(i,j): d_{ij} \text{ определено}\}$ и $N_x=\{(i,j): i<j,\ \bar{d}_{ij} \text{ определено}\}$. Задача локализации датчиков в сети состоит в поиске таких $x^1,..., x^n \in R^d$, которые удовлетворяют следующим условиям:

$$\begin{aligned} \left\|a^i - x^j\right\|^2 &= d_{ij}^2 \quad \forall (i,j)\in N_a, \\ \left\|x^i - x^j\right\|^2 &= \bar{d}_{ij}^2 \quad \forall (i,j)\in N_x. \end{aligned} \tag{1}$$

Таким образом, задача локализации датчиков в сети свелась к решению нелинейной квадратичной системы уравнений (1). Решить эту задачу достаточно сложно и к тому же она может не всегда иметь решение (заданные расстояния между датчиками невозможно реализовать), поэтому заменим ее оптимизационной задачей:

$$\min\left\{\|u\|^2+\|v\|^2 \;\middle|\; \begin{array}{ll} \left\|a^i - x^j\right\|^2 = d_{ij}^2 + u_{ij}, & \forall (i,j),\ i=1,\dots,m,\ j=1,..,k \\ \left\|x^i - x^j\right\|^2 = \bar{d}_{ij}^2 + v_{ij}, & i<j,\ i=1,\dots,k,\ j=2,...,k \end{array}\right\}, \tag{2}$$

где u_{ij} – отклонение от заданного расстояния между i-той вершиной и j-тым датчиком ($\forall (i,j),\ i=1,\dots,m,\ j=1,..,k$), v_{ij} – отклонение от заданного расстояния между i-тым и j-тым датчиками ($i<j,\ i=1,\dots,k,\ j=2,...,k$). Все значения u_{ij}, где $i=1,\dots,m,\ j=1,..,k$, формируют вектор u размерности p, где p – количество известных расстояний d_{ij}. Все значения v_{ij}, где $i<j,\ i=1,\dots,k,\ j=2,...,k$, формируют вектор v размерности l, где l – количество расстояний $\bar{d}_{ij}$.

Таким образом, целевая функция задачи (2) минимизирует отклонения от заданных расстояний. Очевидно, что задача (2) всегда имеет решение.

Эту квадратичную оптимизационную задачу (2) преобразуем к задаче полуопределенной оптимизации. В большинстве ранних алгоритмов данная проблема сводилась к задаче нелинейной глобальной оптимизации. Альтернативный подход с использованием SDP был разработан относительно недавно [11].

3. Полуопределенная релаксация задачи локализации датчиков в сети

Преобразуем задачу (2) к задаче полуопределенной оптимизации (SDP) [4]. Для этого используем преобразование $xAx^T = Axx^T = A \bullet X$, где X – полуопределенная матрица ранга «единица». При решении преобразованной задачи упускается ограничение на ранг матрицы X.

Усовершенствуем релаксацию, которая используется в [11], для нашей постановки задачи (3), учитывающей отклонения от заданных расстояний. Получим следующую задачу полуопределенной оптимизации:

$$\min\left\{ C\cdot X \;\middle|\; \begin{array}{l} \underline{A_{ij}}\cdot X = 0,\ \forall(i,j),\ i=1,\dots,m,\ j=1,...,k,\ X\succeq 0 \\ \underline{A_{ij}}\cdot X = 0,\ \ i<j,\ i=1,\dots,k,\ j=2,...,k,\ X\succeq 0 \end{array} \right\}, \tag{3}$$

где

$$X = \begin{pmatrix} X_1 & X_2 \\ X_2^T & X_3 \end{pmatrix} \text{ размерности } (n+k+p+l)\times(n+k+p+l), \tag{4}$$

$$X_1 = I = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \ddots & 0 \\ & & \ddots & \\ 0 & 0 & \dots & 1 \end{pmatrix} \text{ размерности } n\times n,$$

$$X_2 = \begin{pmatrix} x_1^1 & \dots & x_1^k & u_1 & \dots & u_p & v_1 & \dots & v_l \\ \vdots & \ddots & \vdots & 0 & \dots & 0 & 0 & \dots & 0 \\ x_n^1 & \dots & x_n^k & 0 & \dots & 0 & 0 & \dots & 0 \end{pmatrix} \text{ размерности } n\times(k+p+l),$$

$$X_3 = X_2^T X_2 \text{ размерности } (k+p+l)\times(k+p+l).$$

Очевидно, что матрица C будет иметь следующий вид:

$$C = \begin{pmatrix} 0\dots 0 & 0 \\ \vdots\ \ddots & \vdots \\ 0\dots 0 & 0 \\ 0\dots 0 & I \end{pmatrix}, \tag{5}$$

где I – единичная матрица размерности $(p+l)\times(p+l)$.

Рассчитаем матрицы ограничений A_{ij}. Из (2) имеем

$$\left\|a^i - x^j\right\|^2 - d_{ij}^2 - u_{ij} = -d_{ij}^2 + \left\|a^i\right\|^2 + \left\|x^j\right\|^2 - 2a^i x^j - u_{ij} =$$

$$= -d_{ij}^2 + ({a_1^i}^2 + \dots + {a_n^i}^2) + ({x_1^j}^2 + \dots + {x_n^j}^2) - 2(a_1^i x_1^j + \dots + a_n^i x_n^j) - u_{ij} = 0. \tag{6}$$

Учитывая (4) и (6), имеем

$$A_{ij} = \begin{pmatrix} A_1 & A_2 \\ A_2^T & A_j' \end{pmatrix} \text{ размерности } (n+k+p+l)\times(n+k+p+l), \tag{7}$$

$$A_1 = \begin{pmatrix} \left(-d_{ij}^2 + \left\|a^i\right\|^2\right) & 0\ \dots 0 \\ 0 & 0\ \dots 0 \\ \vdots & \ddots \\ 0 & 0\ \dots\ 0 \end{pmatrix} \text{ размерности } n\times n,$$

$$A_2 = \begin{array}{c} \begin{array}{ccccccc} & & j & & & u_{ij} & \end{array} \\ \begin{pmatrix} 0 & \ldots & -a_1^i & 0 & \ldots & -0.5 & 0 \ldots 0 \\ \vdots & \ddots & \vdots & 0 & \ldots & 0 & 0 \ldots 0 \\ 0 & \ldots & -a_n^i & 0 & \ldots & 0 & 0 \ldots 0 \end{pmatrix} \end{array} \text{ размерности } n \times (k+p+l),$$

$$A'_j = \begin{array}{c} \begin{array}{ccccccc} & & & j & & & \end{array} \\ \begin{pmatrix} 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \ldots & & & & & & \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \ldots & & & & & & \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \end{pmatrix} \end{array} \text{ размерности } (k+p+l) \times (k+p+l).$$

То есть в матрице A_2 в j-том столбце элементы равны компонентам вектора a^i, взятым с противоположным знаком, и элемент, соответствующий u_{ij} в матрице X_2, равен $-0,5$. Все элементы матрицы A'_j равны нулю, кроме $a_{jj} = 1$.

Рассчитаем матрицы ограничений $\overline{A}_{ij}$. Из (2) имеем

$$\left\| x^i - x^j \right\|^2 - \overline{d}_{ij}^2 - v_{ij} = -\overline{d}_{ij}^2 + \left\| x^i \right\|^2 + \left\| x^j \right\|^2 - 2x^i x^j - v_{ij} =$$

$$= -\overline{d}_{ij}^2 + (x_1^{i\,2} + \ldots + x_n^{i\,2}) + (x_1^{j\,2} + \ldots + x_n^{j\,2}) - 2(x_1^i x_1^j + \ldots + x_n^i x_n^j) - v_{ij} = 0. \qquad (8)$$

Тогда, учитывая (4) и (8), имеем

$$\overline{A}_{ij} = \begin{pmatrix} \overline{A}_1 & \overline{A}_2 \\ \overline{A}_2^T & \overline{A}'_{ij} \end{pmatrix} \text{ размерности } (n+k+p+l) \times (n+k+p+l), \qquad (9)$$

$$\overline{A}_1 = \begin{pmatrix} -\overline{d}_{ij}^2 & 0 \ldots 0 \\ 0 & 0 \ldots 0 \\ \vdots & \ddots \\ 0 & 0 \ldots 0 \end{pmatrix} \text{ размерности } n \times n,$$

$$\overline{A}_2 = \begin{array}{c} \begin{array}{ccccccc} & & & \mathrm{v}_{ij} & & & \end{array} \\ \begin{pmatrix} 0 & \ldots & 0 & -0.5 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & 0 & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \end{pmatrix} \end{array} \text{ размерности } n \times (k+p+l),$$

$$\overline{A}'_{ij} = \begin{array}{c} \begin{array}{ccccccc} & & i & & j & & \end{array} \\ \begin{pmatrix} 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \ldots & & & & & & \\ 0 & \ldots & 1 & 0 & -1 & \ldots & 0 \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & \ldots & -1 & 0 & 1 & \ldots & 0 \\ \ldots & & & & & & \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \end{pmatrix} \end{array} \text{ размерности } (k+p+l) \times (k+p+l).$$

То есть в матрице $\overline{A}_2$ все элементы равны нулю, кроме элемента, соответствующего v_{ij} в матрице X_2, который равен $-0,5$. Все элементы матрицы $\overline{A}'_{ij}$ равны нулю, кроме $a_{ii} = a_{jj} = 1$, $a_{ij} = a_{ji} = -1$.

Очевидно, что количество ограничений A_{ij} будет равно p, а количество ограничений $\overline{A}_{ij}$ будет равно l.

Таким образом, задачу (3) можем записать в общем виде:

$$\min\left\{ C \bullet X \;\middle|\; A \bullet X = 0,\; X \succeq 0 \right\}. \tag{10}$$

SDP-алгоритмы предусматривают релаксацию ограничения на ранг матрицы X в задаче (10) (ищется полуопределенная матрица X произвольного ранга). Задача (10) может быть решена алгоритмом внутренней точки [1, 13] или полуопределенным симплекс-методом [14]. Для решения задачи (10) в данной работе использован предложенный авторами полуопределенный симплекс-метод [14].

4. Полуопределенный симлекс-метод для решения задачи SDP

Полуопределенный симплекс-метод [14] ищет решение задачи полуопределенной оптимизации (10) в виде

$$X = \sum \alpha_j X_j\,, \tag{11}$$

где $X_j = x_j x_j^T$ – матрицы ранга «единица», число слагаемых в сумме больше размерности X и $\alpha \geq 0$. Матрицы X_j являются образующими конуса полуопределенных матриц и образуют его начальную аппроксимацию.

Тогда задача полуопределенной оптимизации (10) сводится к задаче линейного программирования:

$$\min\{\sum \alpha_j C \bullet X_j \mid \sum \alpha_j A_i \bullet X_j = b_i, i = 1,...,m, \alpha \geq 0\}. \tag{12}$$

Алгоритм полуопределенного симплекс-метода [14] для задачи (12) на каждой итерации строит базисное решение, преобразуя строку целевой функции (так, чтобы при базисных переменных коэффициенты целевой функции равнялись нулю) и включает в базис столбец с минимальным значением коэффициента целевой функции. При добавлении в сумму (11) новых образующих полуопределенный симплекс-метод рассчитывает строку целевой функции и для этих образующих. Если не существует образующих, для которых оценка в строке целевой функции будет неотрицательной, то решение задачи (12) совпадает с решением задачи (10).

Таким образом, на каждой итерации полуопределенного симплекс-метода решается задача линейного программирования (12) и проверяется полуопределенность матрицы

$$Q = C - \sum C \bullet x_i x_i^T \sum_{j=1}^{m} b_{ij}^{-1} A_j\,,$$

где b_{ij}^{-1} – элементы обратной матрицы базисных элементов. Суммирование проводится по индексам всех базисных столбцов матрицы ограничений задачи (12). Эта матрица совпадает с оценкой целевой функции для новой образующей $x_k x_k^T$. Другими словами, если $x_k^T Q x_k < 0$, то ввод k-го столбца в базис приведет к уменьшению целевой функции задачи (10). Если же матрица Q – полуопределенная, то значение целевой функции задачи (12) не может быть уменьшено, и текущее решение является оптимальным для задачи (10).

Определяем ранг матрицы X. Когда он равен единице, получаем точное решение начальной задачи (2), а если ранг матрицы X больше единицы, то получаем нижнюю оценку решения задачи (2).

Полученная нижняя оценка решения задачи (2) использовалась в качестве начальной точки для нахождения допустимого решении задачи (2) методом внутренней точки [15].

5. Результаты численных экспериментов

Полуопределенный симплекс-метод для решения задачи локализации датчиков в сети был реализован программно средствами VBA для Excel и были проведены значительные численные эксперименты.

В следующем примере задача локализации датчиков в сети с помощью полуопределенной релаксации сводится к задаче полуопределенной оптимизации, которая решалась полуопределенным симплекс-методом [14].

Рассмотрим задачу локализации датчиков в сети с тремя известными вершинами и двумя датчиками. Имеем известные вершины

$$a_1(1,0),\ a_2(2,1),\ a_3(0,3),$$

заданные расстояния между известными вершинами и датчиками

$$d_{11} = d_{21} = d_{31} = d_{12} = 1$$

и расстояние между датчиками

$$\bar{d}_{12} = 1.$$

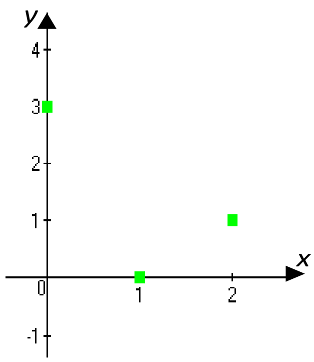

Рис. 1. Заданные вершины

Графически положение известных вершин представлено на рис. 1, из которого очевидно, что задача (1) не имеет решения (невозможно расположить датчики так, чтобы расстояния между ними и известными вершинами были равны единице).

Поэтому будем решать задачу (2), которая имеет следующий вид:

$$\min\left\{ \|u\|^2 + \|v\|^2 \;\middle|\; \begin{array}{ll} \left\|a^i - x^j\right\|^2 = d_{ij}^2 + u_{ij}, & \forall (i,j),\ i = 1,\dots,m,\ \ j = 1,..,k \\ \left\|x^i - x^j\right\|^2 = \bar{d}_{ij}^2 + v_{ij}, & i < j,\ i = 1,\dots,k,\ \ j = 2,...,k \end{array} \right\}. \tag{13}$$

Из условия задачи следует, что $m = 3,\ k = 2,\ n = 2,\ p = 4,\ l = 1$. Преобразуем задачу (13) к задаче полуопределенной оптимизации. Для этого используем схему (3), (4), (5), (7), (9).

Получим следующую задачу полуопределенной оптимизации:

$$\min\left\{ C \cdot X \;\middle|\; \begin{array}{l} A_{ij} \cdot X = 0,\ \ \forall (i,j),\ \ i = 1,\dots,3,\ \ j = 1,...,2,\ \ X \succeq 0 \\ \bar{A}_{12} \cdot X = 0,\ \ X \succeq 0 \end{array} \right\},$$

где

$$C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad A_{11} = \begin{pmatrix} 0 & 0 & -1 & 0 & -0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$A_{21}=\begin{pmatrix} 4 & 0 & -2 & 0 & 0 & -0.5 & 0 & 0 & 0\\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\ -2 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},\quad A_{31}=\begin{pmatrix} 8 & 0 & 0 & 0 & 0 & 0 & -0.5 & 0 & 0\\ 0 & 0 & -3 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$A_{12}=\begin{pmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 & -0.5 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},\quad \bar{A}_{12}=\begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

С помощью полуопределенного симплекс-метода найдено приближенное решение задачи (13), которое равно 5,6485. Матрица решения X имеет ранг 3, а это означает, что мы нашли приближенное решение начальной задачи (13):

$$\begin{matrix} 1,0000 & 0,0000 & 0,7612 & 1,3523 & 1,3914 & 0,8133 & 1,7466 & 0,0001 & 0,0000\\ 0,0000 & 1,0000 & 1,5279 & 0,8366 & 0,0077 & -0,0002 & 0,0009 & 0,0001 & 0,0002\\ 0,7612 & 1,5279 & 2,9138 & 2,3093 & 1,0710 & 0,6188 & 1,3308 & 0,0001 & 0,0002\\ 1,3523 & 0,8366 & 2,3093 & 2,7047 & 1,8888 & 1,0997 & 2,3629 & 0,0001 & 0,0001\\ 1,3914 & 0,0077 & 1,0710 & 1,8888 & 1,9363 & 1,1317 & 2,4303 & 0,0001 & 0,0000\\ 0,8133 & -0,0002 & 0,6188 & 1,0997 & 1,1317 & 0,6615 & 1,4206 & 0,0001 & 0,0000\\ 1,7466 & 0,0009 & 1,3308 & 2,3629 & 2,4303 & 1,4206 & 3,0507 & 0,0001 & -0,0001\\ 0,0001 & 0,0001 & 0,0001 & 0,0001 & 0,0001 & 0,0001 & 0,0001 & 0,0000 & 0,0000\\ 0,0000 & 0,0002 & 0,0002 & 0,0001 & 0,0000 & 0,0000 & -0,0001 & 0,0000 & 0,0000 \end{matrix} \qquad (14)$$

Из матрицы (14) получим приближенное решение (13):

$$x_1=\begin{pmatrix}0,7612\\1,5279\end{pmatrix},\quad x_2=\begin{pmatrix}1,3523\\0,8366\end{pmatrix}$$

и приближенные значения отклонений

$$u_{11}=1,3914,\ u_{12}=0,8133,\ u_{13}=1,7466,\ u_{21}=0,0001,$$
$$v_{12}=0.$$

Для получения верхней оценки используем найденные векторы x, u и v в качестве начальных и будем решать задачу (13) методом внутренней точки [15]. Получим координаты датчиков:

$$x_1=\begin{pmatrix}0,7634\\1,5297\end{pmatrix},\quad x_2=\begin{pmatrix}1,5075\\0,8616\end{pmatrix}$$

при отклонениях

$$u_{11}=1,396,\ u_{12}=0,8097,\ u_{13}=1,7445,\ u_{21}=0,\ v_{12}=0$$

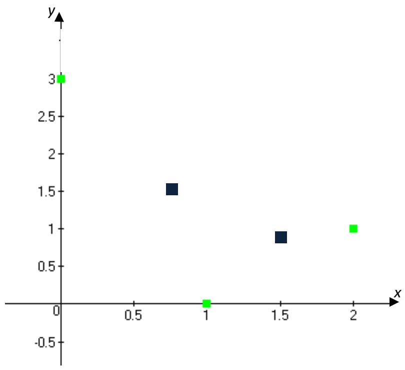

Рис. 2. Оптимальное расположение датчиков

и значение минимума 5,6479, которое является оптимальным.

На рис. 2 изображены расположение вершин и найденное оптимальное расположение датчиков.

6. Выводы

В данной работе решалась задача локализации датчиков в сети путем ее преобразования к общей задаче квадратичной оптимизации. Затем использовалась полуопределенная релаксация. Было проведено значительное количество численных экспериментов, которые подтверждают эффективность выбранного метода для решения задачи локализации датчиков в сети. Приведенная в статье постановка задачи локализации датчиков в сети дает возможность всегда находить решение. Как показали численные эксперименты, использование нижней оценки в качестве начальной точки для нахождения верхней оценки решения задачи в большинстве случаев приводит к нахождению глобального минимума исходной задачи. Таким образом, полуопределенная релаксация в сочетании с методами локального поиска дает возможность находить оптимальное решение задачи локализации датчиков в сети. В дальнейшем авторы видят перспективы использования предложенного подхода для решения других сложных прикладных задач.

СПИСОК ЛИТЕРАТУРЫ

1. Overton M. Semidefinite Programming / M. Overton, H. Wolkowicz // Mathematical Programming. – 1997. – N 77. – P. 105 – 109.
2. Helmberg C. Semidefinite Programming For Combinatorial Optimization / Helmberg C. – Berlin, 2000. – 150 p.
3. Freund R.M. Introduction to Semidefinite Programming / Freund R.M. – Massachusett: Massachusetts Institute of Technology, 2004. – 54 p.
4. Косолап А.І.Використання напіввизначеної оптимізації для моделювання складних систем / А.І. Косолап, А.С. Перетятько // Математичні машини і системи. – 2012. – № 1. – С. 174 – 179.
5. Nie J. Sum of Squares Method for Sensor Network Localization / J. Nie. – University of Minnesota, 2006. – 24 p.
6. Spaseloc: An Adaptive Subproblem Algorithm For Scalable Wireless Sensor Network Localization / Michael W. Carter, Holly H. Jin, M. A. Saunders [et al.] // Siam J. Optim. – 2006. – Vol. 17, N 4. – P. 1102 – 1128.
7. Anderson B.D.O. Wireless sensor network localization techniques / B.D.O. Anderson, G. Mao, B. Fidan // Computer Networks. – 2007. – N 51. – P. 2529 – 2553.
8. Biswas P. Semidefinite programming for ad hoc wireless sensor network localization / P. Biswas, Y. Ye // Proc. of the 3-rd International Symposium on Information Processing in Sensor Networks: Berkeley, CA, USA. – 2004. – P. 46 – 54.
9. Cassioli A. Solving the Sensor Network Localization Problem using an Heuristic Multistage Approach / A. Cassioli. – Universita degli Studi di Firenze, 2009. – 25 p.
10. Krislock N. Explicit Sensor Network Localization using Semidefinite Representations and Facial Reductions / N. Krislock, F. Wolkowicz. – Waterloo: University of Waterloo, 2010. – 35 p.
11. Man-Cho A. Theory of Semidefinite Programming for Sensor Network Localization / A. Man-Cho, Y. Ye. – ACM-SIAM Symposium on Descrete Algorithms, 2004. – 16 p.
12. SFSDP: a Sparse Version of Full Semidefinite Programming Relaxation for Sensor Network Localization Problems / S. Kim, M. Kojima, H. Waki [et al.]. – Tokyo, 2009. – 19 p.
13. Freund R.M. Introduction to Semidefinite Programming / Robert M. Freund. – Massachusett: Massachusetts Institute of Technology, 2004. – 54 p.
14. Косолап А.И. Обобщение симплекс-метода для решения задач полуопределенной оптимизации / А.И. Косолап // Математичне та комп`ютерне моделювання. – 2010. – Вип. 3. – С. 99 – 106.
15. Terlaky T. Interior Point Methods of Mathematical Programming / Terlaky T. – Springer, 1996. – 528 p.

Стаття надійшла до редакції 28.01.2014