Оценка устойчивости решений открытых задач о назначении

Ревотюк М.П.; Кароли М.К.

Кафедра информационных технологий автоматизированных систем

Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

e-mail: rmp@bsuir.by

***Аннотация*—Предложен эффективный алгоритм оценки интервалов устойчивости решений открытых линейных задач о назначении, базирующийся на итерациях реоптимизации текущих решений. Переход к очередной итерации соответствует инверсии вхождения элемента графа задачи в оптимальное паросочетание.**

Ключевые слова: задача о назначении; интервал устойчивости решения; реоптимизация решений

IV. ПОСТАНОВКА ЗАДАЧИ

Решение классических открытых линейных задач о назначении [1], формально записанных в виде

$$\min\left\{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\cdot x_{ij} \;\middle|\; \begin{array}{l} \sum_{i=1}^{m} x_{ij} = \sum_{j=1}^{n} x_{ij} = 1; \\ x_{ij} \ge 0;\ i=\overline{1,m},\ j=\overline{1,n} \end{array}\right\} \qquad (1)$$

есть вектор назначений строк матрицы коэффициентов ее столбцам:

$$R = \left\{ r_j = i \,\middle|\, c_{ij} - u_i - v_j = 0,\ i=\overline{1,m},\ j=\overline{1,n} \right\}. \qquad (2)$$

Здесь $c_{.}$ – элементы матрицы коэффициентов, $u_{.}$ и $v_{.}$ – потенциалы строк и столбцов, определяемые, например, после применения известной процедуры венгерского метода решения задач о назначении [1,2].

Во многих случаях требуется оценка устойчивости назначения. При этом для каждого элемента c_{ij} в (1) необходимо вычислить интервал (s_{ij}, f_{ij}), в котором значения таких элементов могут быть изменены без нарушения существующего назначения (2). Очевидно, что $c_{ij} \in (s_{ij}, f_{ij}),\ i=\overline{1,m},\ j=\overline{1,n}$.

V. СХЕМА РЕШЕНИЯ ЗАДАЧИ

Предлагается определять интервалы устойчивости решения задач вида (1) посредством построения экономной инкрементной схемы реоптимизации текущего решения для каждого элемента матрицы [3].

Можно заметить, что изменение элементов матрицы влечет необходимость пересмотра результата оптимизации, если только меняется позиция нулевого элемента после операции приведения. Действительно, увеличение элемента $c_{ij} \leftarrow c_{ij}^*$ матрицы задачи (1), когда $x_{ij} = 0$, не меняет решения для любых $i \in \overline{1,m}$, $j \in \overline{1,n}$. Аналогично, уменьшение элемента $c_{ij} \leftarrow c_{ij}^*$, когда $x_{ij} = 1$, не нарушает соотношения $c_{ij} = u_i + v_j$ в задаче (1).

В других случаях, как несложно показать, требуется повтор итераций назначения лишь для строк

$$M^* = \left\{ i \,\middle|\, \begin{array}{l} (c_{ij}^* > c_{ij}) \wedge (x_{ij} = 0) \vee \\ (c_{ij}^* < c_{ij}) \wedge (x_{ij} = 1),\ j=\overline{1,n} \end{array} \right\},$$

что влечет, как легко заметить, снижение вычислительной сложности пересчета задачи (1) на величину $\left(m - \left|M^*\right|\right)\cdot O(n^2)$.

Изменение элементов матрицы должно отражаться значениями потенциалов, сбросом признака назначения соответствующей дуги графа и итерацией назначения строки. Если изменены элементы строки i', то ее потенциал должен быть скорректирован:

$$u_{i'} = \min_j (c_{i'j} - v_j,\ j=\overline{1,n}).$$

В случае изменения элементов столбца j' его потенциал также должен меняться:

$$v_{j'} = \min_i (c_{ij'} - u_i,\ i=\overline{1,m}).$$

При этом элементы матрицы разбиваются на два множества, учитывая их вхождение в сумму целевой функции (1). Для каждого из таких множеств применяется отдельная процедура реоптимизации.

VI. ОЦЕНКА ГРАНИЦ ИНТЕРВАЛОВ

Используя элементы решения в виде (2), легко выделить ребра графа совершенного паросочетания:

$$E_m = \left\{ (r_j, j) \,\middle|\, (r_j < m),\ j=\overline{1,n} \right\}.$$

Интервал значений веса любого ребра такого графа, когда назначение остается неизменным, может быть описан как $(s_{ij}, f_{ij})_m = (-\infty, c_{ij} + \Delta_{ij}^m], (i,j) \in E_m$. Последнее означает, что для задачи минимизации (1) существующий вес назначенных ребер можно увеличить без нарушения структуры текущего решения на величину $\Delta_{ij}^m, (i,j) \in E_m$. Превышение такой величины приведет к скрытию соответствующего ребра. Отсюда очевидно следует схема алгоритма определения $\Delta_{ij}^m, (i,j) \in E_m$.

Пусть оценка оптимального назначения есть Z^0. Очевидно, что если положить $c_{xy} = \infty$, то ребро $x \to y$ будет скрыто.

Реоптимизация решения может быть проведена относительно строки x. Изменение потенциалов такой строки составит $u_x^m - u_x^0 = Z_{xy}^m - Z^0$ [4]. Здесь Z_{xy}^m – оценка нового решения без включения ребра $x \to y$. Действительно, скрытие ребра $x \to y$ не влияет на значения потенциалов других строк. Процесс реоптимизации, начинающийся в вершине x, завершится в вершине y, потенциал которой тоже не изменится [1]. Меняется только потенциал u_x, поэтому $\Delta_{xy}^m = Z_{xy}^m - Z^0$.

Ранее приведенные рассуждения можно провести и для элементов вне оптимального совершенного паросочетания

$$E_u = \{(i, j),\ i = \overline{1, m},\ j = \overline{1, n}\} \setminus E_m,$$

которые представляют множество скрытых ребер графа решения задачи (1). Интервал значений веса любого скрытого ребра, для которого оптимальное назначение остается неизменным, может быть описан как $(s_{ij}, f_{ij})_u = [c_{ij} - \Delta_{ij}^u,\ +\infty), (i, j) \in E_u$.

Последнее означает, что существующий вес для скрытых ребер можно увеличить без нарушения структуры текущего решения задачи минимизации. Ребро графа решения перестанет быть скрытым после назначения веса из интервала $(-\infty,\ c_{ij} - \Delta_{ij}^u)$. Таким образом, выполнив процедуру реоптимизации после фиксации $c_{xy} = -\infty$, становится известным значение Z_{xy}^u оценки решения с ребром $x \to y$. В результате получаем $\Delta_{xy}^u = Z_{xy}^u - Z^0$.

VII. ПРИМЕР ОЦЕНКИ УСТОЙЧИВОСТИ

Рассмотрим пример решения задачи о назначении с оценкой интервалов устойчивости решения. Пусть матрица коэффициентов задачи (1) имеет вид

$$C = \begin{pmatrix} 87 & 62 & 64 & 43 & 72 & \underline{57} \\ \underline{52} & 68 & 94 & 63 & 76 & 83 \\ 77 & \underline{48} & 64 & 54 & 59 & 68 \\ 58 & 49 & 71 & \underline{42} & 66 & 89 \end{pmatrix}.$$

Результат решения задачи отражен подчеркнутыми элементами матрицы, а вектор решения в форме (2):

$$R = \begin{pmatrix} 2 & 3 & 5 & 4 & 5 & 1 \end{pmatrix}.$$

Здесь матрица коэффициентов имеет параметры $m = 4$, $n = 6$. Элементы вектора решения в форме (2), для которых $r_j > m,\ j \in \overline{1, n}$, соответствуют столбцам, не попавшим в оптимальное назначение.

Результат построения матрицы начальных границ интервалов устойчивости:

$$S = \begin{pmatrix} 33 & 46 & 57 & 39 & 57 & -\infty \\ -\infty & 41 & 52 & 36 & 52 & 50 \\ 32 & -\infty & 55 & 34 & 55 & 55 \\ 32 & 45 & 56 & -\infty & 56 & 56 \end{pmatrix}$$

Результат построения матрицы конечных границ интервалов устойчивости:

$$F = \begin{pmatrix} +\infty & +\infty & +\infty & +\infty & +\infty & 61 \\ 76 & +\infty & +\infty & +\infty & +\infty & +\infty \\ +\infty & 52 & +\infty & +\infty & +\infty & +\infty \\ +\infty & +\infty & +\infty & 46 & +\infty & +\infty \end{pmatrix}$$

Очевидно, что тройки $(s_{ij}, c_{ij}, f_{ij}),\ i = \overline{1, m},\ j = \overline{1, n}$ характеризуют устойчивость текущего назначения.

VIII. ЗАКЛЮЧЕНИЕ

Предлагаемый алгоритм оценки интервалов устойчивости задач о назначении характеризуется вычислительной сложностью $O(m^2 n)$ для открытых задач с прямоугольными матрицами исходных данных, когда $m \le n$.

Случай закрытых задач о назначении, когда $m = n$, рассмотрен в [4] без использования реоптимизации с оценкой вычислительной сложности $O(n^4)$. Принципиальное отличие предложенного здесь подхода – использование всей предыстории процесса оптимизации, что влечет снижение сложности оценки локального изменения целевой функции после скрытия ребра.

Случай открытых задач о назначении, для которых $m > n$, разрешается тривиальным транспонированием матрицы исходных данных.

Рассмотренный подход может быть применен и для задач о назначении по критерию максимума целевой функции (1). Для этого достаточно инверсии знака элементов матрицы и зеркального отображения границ интервалов.

[1] Jonker R., Volgenant A. A shortest path algorithm for dense and sparse linear assignment problem//Computing, vol. 38, 1987. – pp. 325-340.

[2] Toroslu I.H., Üçoluk G. Incremental assignment problem// Information Sciences, vol. 177, 2007. – pp. 1523-1529.

[3] Ревотюк, М.П. Реоптимизация решения задач о назначении/ М.П. Ревотюк, П.М. Батура, А.М. Полоневич//Доклады БГУИР. – 2011. – № 1(55). – С. 55-62.

[4] Lantao, Liu. Assessing optimal assignment under uncertainty: An interval-based algorithm/Lantao Liu, Dylan A Shell//The International Journal of Robotics Research, 2011. Vol. 30(7). – pp. 936-953.