УДК 519.112

АЛГОРИТМИЗАЦИЯ ОДНОЙ ЗАДАЧИ ПЕРЕДАЧИ ИНФОРМАЦИИ

© 2003 И. В. Фёдорова, Т. В. Фёдорова, Г. Д. Чернышова

Воронежский государственный университет

Рассматривается задача дискретной оптимизации, возникающая при исследовании проблемы передачи информации. Предлагаются алгоритмы решения, основанные на теории двойственности и на возможности оценивания исходной постановки задачей более простой структуры.

Рассматривается задача дискретной оптимизации с нелинейной функцией и ограничениями транспортного типа:

$$\sum_{j=1}^{n} \operatorname{sgn}\left(\sum_{i\in I_j} x_{ij}\right) \to \min, \tag{1}$$

$$\sum_{j\in J_i} x_{ij} = 1, \; i=\overline{1,m}, \tag{2}$$

$$\sum_{i\in I_j} x_{ij} \le D_j, \; j=\overline{1,n}, \tag{3}$$

$$x_{ij} = \{0,1\}, \; i=\overline{1,m}, \; j=\overline{1,n}. \tag{4}$$

Множества I_j, J_i формируются на основе булевой матрицы A :

$I_j = \{i : a_{ij} = 1\}$ $j=\overline{1,n}$; $J_i = \{j : a_{ij} = 1\}$, $i=\overline{1,m}$.

$$\operatorname{sgn}\left(\sum_{i\in I_j} x_{ij}\right) = 1, \text{ если } \left(\sum_{i\in I_j} x_{ij}\right) > 0;$$

$$\operatorname{sgn}\left(\sum_{i\in I_j} x_{ij}\right) = 0, \text{ если } \left(\sum_{i\in I_j} x_{ij}\right) = 0.$$

Эта задача возникает, например, при размещении на местности опорных пунктов передачи радиосигнала. Требуется выбрать минимальное число таких опорных пунктов, при условии получения сигнала каждым объектом-абонентом. При этом существуют ограничения на количество обслуживаемых объектов для каждого опорного пункта. Считается, что j-й пункт может обслуживать i-й объект, если последний находится в радиусе его действия. Исходные данные задачи можно представить в виде булевой матрицы $A = (a_{ij})$, $i=\overline{1,m}$, $j=\overline{1,n}$:

$$a_{ij} = \begin{cases} 1, \text{ если j - й опорный пункт может} \\ \quad \text{обслуживать i - й объект.} \\ 0, \text{ иначе.} \end{cases}$$

Переменные в этом случае вводятся следующим образом:

$$x_{ij} = \begin{cases} 1, \text{ если j - й опорный пункт назначается} \\ \quad \text{для обслуживания i - го объекта } (a_{ij} \ne 0). \\ 0, \text{ иначе.} \end{cases}$$

$$i=\overline{1,m}, \; j=\overline{1,n}.$$

Для приближенного решения задачи (1)—(4) можно использовать двойственные алгоритмы, алгоритмы «жадного» типа и адаптивные алгоритмы, позволяющие оценивать среднее значение функции.

На каждом этапе алгоритмов «жадного» типа делается попытка максимального достижения некоторой локальной цели в соответствии с выбранным критерием. Для оценки предыдущих и последующих шагов алгоритма и повышения результативности его работы могут быть использованы дополнительные эвристические приемы.

Рассмотрим два «жадных» алгоритма для задачи (1)—(4):

Алгоритм 1. Исходными данными для алгоритма являются матрица A и D_j. Заметим, что если $\sum_{i\in I_j} a_{ij} > D_j$, то ограничение (3) несущественно и можно считать, что $D_j = \sum_{i\in I_j} a_{ij}$.

Идея данного алгоритма заключается в выборе столбцов матрицы A по некоторым признакам.

Будем говорить, что столбец j покрывает строку i, если $a_{ij} = 1$. Обозначим через Cov множество выбранных столбцов матрицы A, а через R множество покрытых строк (первоначально $Cov = \{\varnothing\}$, $R = \{\varnothing\}$). Тогда на первом этапе алгоритма сначала рассматриваются все столбцы с максимальным D_j. Если такой столбец один, то он сразу включается в Cov, а если их несколько, то выбирается столбец, покрывающий максимальное число строк. Затем в Cov последовательно включаются столбцы, на которых достигается $\max\limits_{j\notin Cov}\left[\min\left\{D_j, \sum\limits_{i\notin R} a_{ij}\right\}\right]$, при этом при включении в Cov каждого нового столбца соответственно изменяется множество R. Первый этап заканчивается, когда покрыты все m строк.

На втором этапе алгоритма вычисляется величина $def = m - \sum\limits_{j\in Cov} D_j$. Если $def > 0$, то формируется множество Row, мощности def, состоящее из номеров первых def строк, которые покрываются минимальным количеством столбцов из Cov. В множество Cov включается столбец, на котором достигается $\max\limits_{j\notin Cov}\left[\left\{D_j, \sum\limits_{i\in Row} a_{ij}\right\}\right]$. При включении нового столбца в множество Cov значение def уменьшается на соответствующее ему D_j. На данном этапе процесс выбора столбца для включения в Cov продолжается до тех пор, пока значение def не станет меньше либо равно 0. Затем осуществляется попытка построения допустимой в задаче (1)—(4) точки x_{ij}, которая и является приближенным решением задачи (1)—(4). Точка x_{ij} строится следующим образом: в выбранных в результате работы приведенного алгоритма столбцах $j \in Cov$ матрицы $X = (x_{ij})$ с помощью алгоритма с возвратом делается попытка расставить m единиц (в каждой строке по единице) [1].

Алгоритмическая схема данного алгоритма имеет следующий вид:

Шаг 1. Задаются A, D_j. Полагаем $Cov := \{\varnothing\}$, $R := \{\varnothing\}$, $D_j := \min\{D_j |I_j|\}$.

Шаг 2. Формируется множество $S = \{k : \max\limits_{j=\overline{1,n}} D_j = D_k, k = \overline{1,n}\}$ и выбирается номер $I \in S : \max\limits_{j\in S} | I_J | = I_l$. $Cov = Cov \cup 1$, $R = R \cup \{i : a_{il} = 1,\ i = \overline{1,m}\}$.

Шаг 3. $\forall j \notin \mathrm{Cov}$ вычисляется $count[j] = \min\left\{D_j, \sum\limits_{i\notin R} a_{ij}\right\}$ и выбирается номер $l : \max\limits_{j\notin Cov} count[j] = count[l]$. $Cov = Cov \cup I$, $R = R \cup \{i : a_{il} = 1, i = \overline{1,m}\}$.

Шаг 4. Если $|R| \le m$, то переход к шагу 3, иначе — переход к шагу 5.

Шаг 5. Вычисляется $def = m - \sum\limits_{j\in Cov} D_j$.

Шаг 6. Если $def \le 0$, то строится допустимая точка и переход к пункту 10. Иначе переход к шагу 7.

Шаг 7. $\forall i \notin T$ вычисляются $rcount[i] = \sum\limits_{j\in Cov} a_{ij}$, где $T = \{i : \sum\limits_{j\notin Cov} a_{ij} = 0\}$. Отбирается def номеров строк, соответствующих первым def элементам $rcount[i]$, $i \notin T$, расположенным по возрастанию:

$$rcount[i_1] \le rcount[i_2] \le \ldots \le rcount[i_{def}].$$

$$Row = \{i_1 \ldots i_{def}\}.$$

Шаг 8. $\forall j \notin Cov$ вычисляется $count[j] = \min\{D_j, \sum\limits_{i\in Row} a_{ij}\}$ и выбирается номер $l : \max\limits_{j\notin Cov} count[j] = count[l]$. $Cov = Cov \cup l$, $def := def - D_l$. $Row := \{\varnothing\}$.

Шаг 9. Если $def \le 0$, то строится допустимая точка и переход к пункту 10. Иначе переход к пункту 7.

Шаг 10. Останов.

Алгоритм 2. Первоначальной основой для построения следующего алгоритма стала эквивалентная запись задачи (1)—(4) в виде [2]:

$$f(x) = \sum_{j=1}^{n} \mathrm{sgn}(\sum_{i\in I_j} x_{ij}) - \min\{m,n\}\sum_{i=1}^{m}\chi_i(\sum_{j\in J_i} x_{ij}) \to \min_{x\in\Omega_1}, \tag{5}$$

где $\chi(\alpha) = 1$, если $\alpha = 1$ и $\chi(\alpha) = 0$ в противном случае;

$$\Omega_1 = \{x : \sum_{i\in I_j} x_{ij} \le D_j\ j = \overline{1,n}, x_{ij} \in \{0;1\} i = \overline{1,m}\ j \in J_i\}.$$

Вероятностная постановка задачи (5) имеет вид:

$$M_X[f(X)] \to \min_{\{X\}:\ x\in\Omega_1} \qquad (6),$$

где M — операция математического ожидания, $\{X\}$ — множество случайных булевых векторов, реализации которых принадлежат множеству Ω_1.

Вводятся следующие обозначения: $P(X_{ij} = 1) = p_{ij}, P(X_{ij} = 0) = 1 - p_{ij} = q_{ij}$.

Математическое ожидание в случае данной функции может быть вычислено в явном виде:

$$M_X[f(X)] =$$

$$= n - \sum_{j=1}^{n} \prod_{i\in I_j} q_{ij} - \min\{m,n\}\sum_{i=1}^{m}\sum_{k\in J_i}\prod_{j\in J_i\setminus\{k\}}(1-q_{ik})q_{ij}$$

Рассматривается приближенный алгоритм решения задачи (1)—(4), использующий вероятностную постановку (6) для оценивания средних значений функции на каждом шаге. Проверка выполнения условия $X : x \in \Omega_1$ осуществляется отдельно на одном из этапов работы алгоритма.

Алгоритмические схемы решения задачи вида (6) включают в себя следующие этапы:

1. Задание движения на множестве случайных величин $\{X\}$.

2. Отыскание неизвестных параметров движения на основе условия локального улучшения (УЛУ):

$$M[f(X^{N+1})] - M[f(X^N)] \le 0. \qquad (7)$$

3. Итеративный пересчет условных вероятностей.

4. Изменение безусловных вероятностей.

5. Фиксирование допустимых точек в соответствии с полученным распределением.

6. Проверка на останов. Переход к пункту 2.

Воспользуемся этой схемой для построения приближенного алгоритма решения задачи (1)—(4).

Зададим движение на множестве случайных булевых векторов X в виде: $X^{N+1} = \bar{U}^{N+1}X^N + U^{N+1}Y^{N+1}$.

Здесь Y — случайный булев вектор, задающий движение на $N+1$ шаге, U — булева случайная величина.

Обозначим $P\{Y_{ij}^{N+1} = 1\} = \pi_{ij}^{N+1}$ $i = \overline{1,m}, j \in J_i$; $P\{U^{N+1} = 1\} = p_u^{N+1}$,, тогда формула движения в вероятностных характеристиках будет иметь вид:

$$p_{ij}^{N+1} = q_u^{N+1}p_{ij}^N + p_u^{N+1}\pi_{ij}^{N+1} \qquad (8).$$

В предположении о независимости случайных векторов X^N и Y^{N+1}, УЛУ может быть переписано следующим образом:

$$(\pi_{kl}^{N+1} - p_{kl}^N)(\prod_{\substack{i\in I_l\\ i\ne k}} q_{il}^N - \min\{m,n\}\prod_{\substack{j\in J_k\\ j\ne l}} q_{kj}^N +$$

$$+\min\{m,n\}\sum_{\substack{t\in J_k\\ t\ne l}}(1-q_{kt}^N)\prod_{j\in J_k\setminus\{t;l\}} q_{kj}^N) \le 0$$

Введем обозначения:

$$B_{kl} = \prod_{\substack{i\in I_l\\ i\notin k}} q_{il}^N - \min\{m,n\}\prod_{\substack{j\in J_k\\ j\ne l}} q_{kj}^N +$$

$$+\min\{m,n\}\sum_{\substack{t\in J_k\\ t\ne l}}(1-q_{kt}^N)\prod_{j\in J_k\setminus\{t;l\}} q_{kj}^N.$$

Тогда для решения последнего неравенства достаточно положить, например, $\pi_{kl}^{N+1} = 1$, если $B_{kl} \le 0$ и $\pi_{kl}^{N+1} = 0$,если $B_{kl} \ge 0$.

Различные способы выбора пары номеров (k, l) на каждом шаге алгоритма порождают различные алгоритмические версии. Так, например, если пары (k, l) меняются соответственно начальному зафиксированному порядку, то в итоге получим вероятностный аналог алгоритма покоординатного спуска.

Рассматриваемый далее приближенный алгоритм является двойственным и основан на решении задачи вида (9):

$$\begin{gathered}\sum_{j=1}^{n} x_j \to \min\\ \sum_{j=1}^{n} a_{ij}x_j \ge 1, \forall i = \overline{1,m}\\ \sum_{j=1}^{n} b_j x_j \ge 1, b_j \min(D_j, |I_j|/m)\\ x_j \in \{0,1\}, j = \overline{1,n}\end{gathered} \qquad (9)$$

Можно показать, что задача (9) обладает следующим свойством:

Если x_{ij} — допустимая точка задачи (1)—(4), то $x_j = \mathrm{sgn}(\sum_{i\in I_j} x_{ij})$ — допустимая точка задачи (9).

Доказательство.

Пусть x_{ij} допустима в (1—4). Тогда $\sum_{j\in J_i} x_{ij} = 1$, $i = \overline{1,m} \Rightarrow$. $\forall i = \overline{1,m}$ $\exists$ номер $s_i \in J_i : x_{is_i} = 1$. Так как $s_i \in J_i$, то $i \in I_{s_i} \Rightarrow x_{s_i}\,\mathrm{sgn}(\sum_{k\in I_{s_i}\setminus i} x_{ks_i} + 1) = 1$ и

$\sum_{j=1}^{n} a_{ij}x_j = \sum_{j\in J_i} x_j = \sum_{j\in J_i\setminus s_i} x_j + 1 \ge 1$. Так как x_{ij} допустима в (1)—(4), то $\sum_{i=1}^{m}\sum_{j\in J_i} x_{ij} = m$ и $\sum_{i\in I_j} x_{ij} \le D_j$.

Заметим, что $\sum_{i\in I_j} x_{ij} \le D_j \Rightarrow \sum_{i\in I_j} x_{ij} \le$ $\le \min(D_j, |I_j|)\,\mathrm{sgn}(\sum_{i\in I_j} x_{ij})$. Тогда $m = \sum_{i=1}^{m}\sum_{j\in J_i} x_{ij} =$

$= \sum_{j=1}^{n} \sum_{i\in I_j} x_{ij} \le \sum_{j=1}^{n} \min(D_j, |I_j|) \operatorname{sgn}(\sum_{i\in I_j} x_{ij}) \le \sum_{j=1}^{n} \min(D_j, |I_j|) x_j$.

То есть $\sum_{j=1}^{n} b_j x_j \ge 1$, $b_j = \min(D_j, |I_j|)/m$. Таким образом, x_j — допустимая точка задачи (9).

Будем использовать следующее обозначение: $S = \{x : \sum_{j=1}^{n} b_j x_j \ge 1, x_j \in \{0,1\}\}$.

Перепишем задачу (9), используя функцию Лагранжа:

$$\min_{x\in S} \max_{y\ge 0} [\sum_{j=1}^{n} x_j + \sum_{i=1}^{m} y_i (1 - \sum_{j=1}^{n} a_{ij} x_j)] =$$
$$= \min_{x\in S} \max_{y\ge 0} [\sum_{j=1}^{n} (1 - \sum_{i=1}^{m} y_i a_{ij}) x_j \sum_{i=1}^{m} y_i.$$

Двойственной к ней является задача:

$$\max_{y\ge 0} \min_{x\in S} \left[\sum_{j=1}^{n} (1 - \sum_{i\in I_j} y_i) x_j \sum_{i=1}^{m} y_i \right] \quad (10)$$

Для отыскания решения задачи (10) применяется двойственный метод Удзавы. На этапе минимизации по x при фиксированном $\hat{y}$ решается задача (11):

$$\sum_{j=1}^{n} c_j x_j \to \min$$
$$\sum_{j=1}^{n} b_j x_j \ge 1 \quad (11)$$
$$x_j \in \{0,1\}, j = \overline{1,n} \text{ где } c_j = 1 - \sum_{i\in I_j} \hat{y}_i$$

Для решения (11) могут использоваться приближенные алгоритмы решения задачи о ранце [3].

Другие двойственные алгоритмы рассмотрены в [4].

Двойственный адаптивный алгоритм основан на вероятностной интерпретации множителей Лагранжа [5] и предполагает адаптивный учет ограничений для задачи (12), которая эквивалентна задаче (1)—(4):

$$f(x) = \sum_{j=1}^{n} \operatorname{sgn}(\sum_{i\in I_j} x_{ij}) - \min\{m,n\} \sum_{i=1}^{m} \chi_i (\sum_{j\in J_i} x_{ij}) \to \min$$
$$g_j = \sum_{i\in I_j} x_{ij} \le D_j, j = \overline{1,n}$$
$$x_{ij} \in \{0,1\}, i = \overline{1,m}, j = \overline{1,n}, \quad (12)$$

После нормирования множителей Лагранжа, функция Лагранжа для задачи (12) принимает следующий вид:

$$L(x,y) =$$
$$= (1 + \sum_{j=1}^{n} y_j) \left[\frac{1}{1 + \sum_{j=1}^{n} y_j} f(x) + \sum_{j=1}^{n} \frac{y_j}{1 + \sum_{j=1}^{n} y_j} g_j(x) \right].$$

Если ввести обозначения: $p_0 = \dfrac{1}{1 + \sum_{j=1}^{n} y_j}$, $p_j = \dfrac{y_j}{1 + \sum_{j=1}^{n} y_j}$, $j = \overline{1,n}$, то тогда функция Лагранжа переписывается в виде:

$$L(x,y) = \frac{1}{p_0} \left[p_0 f(x) + \sum_{j=1}^{n} p_j g_j(x) \right].$$

Заметим, что $0 \le p_j \le 1$, $j = \overline{0,n}$ и, кроме того, $\sum_{j=0}^{n} p_j = 1$. Таким образом, величины p_j можно трактовать как вероятности некоторых случайных величин ξ_j, таких, что $\xi_j = 1$ с вероятностью p_j и $\xi_j = 0$ с вероятностью $q_j = 1 - p_j$, причем $\sum_{j=0}^{n} \xi_j^{(r)} = 1$, где $\xi_j^{(r)}$ – реализация случайной величины ξ_j.

Используя введенные вероятностные характеристики, задачу (12) можно переписать в терминах математического ожидания в виде:

$$\min_{x_{ij}\in\{0,1\}} \max_{0\le p_j \le 1} \frac{1}{p_0} M_\xi \left[\xi_0 f(x) + \sum_{j=1}^{n} \xi_j g_j(x) \right], \quad (13)$$

где $\xi = (\xi_0, \xi_1, \ldots, \xi_n)$.

Для поиска решения задачи (13) предлагается общая модельная схема, состоящая из 5 этапов:

1. Инициализация.
2. Отыскание неизвестных параметров движения согласно условию локального улучшения (УЛУ) при фиксированном x^N. Изменение вероятностей p_0^N, p_1^N,... , p_n^N по заданной формуле движения с учётом найденных параметров.
3. Анализ полученного распределения и формирование поисковой функции $F(x,p)$.
4. Минимизации поисковой функции по x.
5. Проверка условий останова.

Рассмотрим отдельные этапы модельной схемы:

2. Рассмотрим отдельную задачу

$$\frac{1}{p_0} M_\xi \left[\xi_0 f(x) + \sum_{j=1}^{n} \xi_j g_j(\hat{x}) \right] \to \max_{0 \le p_j \le 1} \quad (14)$$

при фиксированном векторе $\hat{x}$.

Движение на множестве случайных величин ξ_j задается следующим образом:

$$\xi_j^{N+1} = (1 - u^{N+1})\xi_j^N + u^{N+1}\eta_j^{N+1}, j = \overline{0,n}, \quad (15)$$

где N — номер итерации, u^{N+1}, η_j^{N+1} — булевы случайные величины, неизвестные параметры движения, определяемые на основе условия локального улучшения:

$$\frac{1}{p_0^{N+1}} M_{\xi^{N+1}} \left[\xi_0^{N+1} f(\hat{x}) - \sum_{j=1}^{n} \xi_j^{N+1} g_j(\hat{x}) \right] - \\ - \frac{1}{p_0^N} M_{\xi^N} \left[\xi_0^N f(\hat{x}) - \sum_{j=1}^{n} \xi_j^N g_j(\hat{x}) \right] \ge 0. \quad (16)$$

Введем обозначения: $P(\eta_j^N = 1) = \pi_j^N$, $j = \overline{0,n}$, $P(u^{N+1} = 1) = p_u^{N+1}$, $P(u^{N+1} = 0) = 1 - p_u^{N+1} = q_u^{N+1}$.

Тогда формула движения (15) в вероятностных характеристиках будет иметь вид:

$$p_j^{N+1} = q_u^{N+1} p_j^N + p_u^{N+1} \pi_j^{N+1}, j = \overline{0,n}. \quad (17)$$

Если выбрать номер l и положить $\pi_j^{N+1} = p_j^N$, $\forall\ j \notin \{0, l\}$, то УЛУ после ряда преобразований примет вид: $(\pi_l^{N+1} - p_l^N)W \ge 0$, где

$$W = p_0^N g_l(\hat{x}) + \sum_{j=1}^{n} p_j^N g_j(\hat{x}). \quad (18)$$

Возможны различные варианты формирования константы W в соответствии с распределением вероятностей p_j^N, $j = \overline{1,n}$.

В качестве одного из решений неравенства (18) можно взять, например, $\pi_l^{N+1} = p_l^N + \lambda W$, где $\lambda \ge 0$ — параметр, обеспечивающий выполнение условия $0 \le \pi_l^{N+1} \le 1$.

Тогда формулы пересчета вероятностей принимают следующий вид:

$$p_l^{N+1} = p_l^N + p_u^{N+1} \lambda W, \quad (19)$$

$$p_0^{N+1} = p_0^N - p_u^{N+1} \lambda W, \quad (20)$$

где $\lambda W \in [0; p_0^N]$, при $W > 0$; $\lambda W \in [-p_l^N; 0]$, при $W < 0$; $\lambda W = 0$, при $W = 0$.

3. В качестве “поисковой” функции выбирается

$$\mathrm{F}(x, p^N) = p_0^N f(x) + \sum_{j \in J} p_j^N g_j(x), \quad (21)$$

где $J = \{ j : p_j^N > \underline{p}^N \}$, $\underline{p}^N$ — настраиваемое нижнее пороговое значение.

5. В данном алгоритме используются стандартные критерии останова: по максимальному числу итераций, по числу итераций после последней смены рекорда, по близости значений функции в последовательно полученных точках.

ЛИТЕРАТУРА

1. *Вирт Н.* Алгоритмы и структуры данных / Пер. с англ. Д. Б. Подшивалова. — М.: Мир, 1989. — 360 с.

2. *Каширина И. Л.* Эквивалентные преобразования одной задачи транспортного типа, позволяющие использовать различные методы ее решения. / И. Л. Каширина, Г. Д. Чернышова // Вестник ВГУ, серия «Физика, математика», Воронеж, 2001, № 2. — С. 104—107.

3. *Алексеев О. Г.* Комплексное применение методов дискретной оптимизации / М.: Наука. Гл. ред. физ-мат. лит., 1987. —248 с.

4. *Мухин А. В.* Алгоритмический комплекс для решения задачи выбора ретрансляторов. / А. В. Мухин, И. В. Фёдорова, Т. В. Фёдорова, Г. Д. Чернышова // Вестник ВГТУ, серия «Вычислительные и информационно-телекоммуникационные системы», вып. 8.2, Воронеж, 2002. С. 36—39.

5. *Каплинский А. И.* Адаптивный выбор ограничений в задаче оптимального проектирования. / А. И. Каплинский, Г. Д. Чернышова // Оптимизация и проектирование человеко-машинных систем (сборник научных трудов), ВПИ, Воронеж, 1980, С. 56—60.