ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Январь—июнь 2001. Серия 2. Том 8, № 1, 70–87

УДК 519.854.2

ЗАДАЧА РАЗМЕЩЕНИЯ ПРЯМОУГОЛЬНЫХ ОБЪЕКТОВ С МИНИМАЛЬНОЙ СТОИМОСТЬЮ СВЯЗЫВАЮЩЕЙ СЕТИ

А. В. Панюков

Предложен способ иерархической декомпозиции задачи размещения прямоугольных объектов с минимальной стоимостью связывающей их сети на задачу оптимального упорядочения (верхний уровень) и двух задач построения оптимального потока (нижний уровень). Получены следующие результаты: 1) найдены необходимые и достаточные условия локального экстремума и предложен алгоритм построения локально-оптимальных решений; 2) для задач большой размерности предложен алгоритм решения, основанный на случайном поиске, эвристике и рассмотренном методе декомпозиции; 3) для поиска глобального экстремума предложен алгоритм по схеме метода ветвей и границ.

Введение

Задача размещения прямоугольных объектов является математической моделью многих задач управления и проектирования. Формальная постановка задачи заключается в нахождении для заданного множества J объектов и симметричных неотрицательных функций $c, r, d : J^2 \to \mathbf{Z}^{+2}$, такого отображения (способа размещения) $\psi = (\psi_1, \psi_2) : J \to \mathbf{Z}^2$, при котором достигает минимума функционал обобщенной стоимости связывающей сети

$$F(\psi) = \sum_{i=1}^{2} \sum_{l,k \in J} c_i\{l,k\} |\psi_i(l) - \psi_i(k)| \to \min_{\psi \in \Psi}, \tag{1}$$

а множество Ψ допустимых размещений определяется следующей системой ограничений:

а) соблюдение технологических разрывов между границами размещаемых объектов

$$(\forall l, k \in J)\Big[\max_{i \in \{1,2\}} \{|\psi_i(l) - \psi_i(k)| - r_i\{l,k\}\} \geqslant 0\Big]; \tag{2}$$

б) ограниченность максимально допустимых расстояний вдоль каждой координатной оси

$$(\forall l, k \in J,\ \forall i \in \{1,2\})[|\psi_i(l) - \psi_i(k)| \leqslant d_i\{l,k\}]. \tag{3}$$

Во многих практических задачах, в частности в задачах управления и планирования, добавляется требование частичной упорядоченности вдоль координатных осей в соответствии с заданными отношениями частичного порядка R_1^0 и R_2^0 на множестве J

$$(\forall (l,k) \in R_i,\ \forall i \in \{1,2\})[\psi_i(l) \geqslant \psi_i(k)]. \tag{4}$$

Как будет видно из дальнейшего, данное ограничение не вносит принципиальных трудностей. Для большей лаконичности изложения оно опускается и упоминается лишь в комментариях к полученным результатам.

В основе большей части известных эвристических, стохастических и интерактивных алгоритмов размещения прямоугольных объектов лежит метод последовательно-одиночного размещения [15, 19], заключающийся в последовательном определении позиций размещаемых элементов при условии, что уже размещенные элементы считаются неподвижными. В случае размещения разногабаритных объектов наиболее известен метод последовательно-одиночного размещения, использующий понятие годографа вектор-функции плотного размещения [19, 20].

Основным недостатком метода последовательно-одиночного размещения является недостаточное качество построенных с его помощью решений, в частности из-за неудачного выбора последовательности размещения объектов и из-за жесткой фиксации ранее размещенных объектов. Первая из указанных причин устраняется методом последовательно-одиночного размещения в совокупности с методами оптимизации на перестановках, определяющих последовательность размещения объектов [19, 20]. Данный подход, как сообщается в [20], позволяет строить достаточно хорошие решения, однако остается невозможность гарантированного построения локального экстремума задачи.

Точные алгоритмы решения задачи размещения прямоугольных объектов используют формализацию данной задачи в виде задачи целочисленного линейного программирования [2, 3]. В работе [2] сообщается о том, что осуществлена программная реализация решения этой задачи методом ветвей и границ и методом расчленения Бендерса [21]. Комбинаторный алгоритм [2] использует лексикографическое упорядочивание объектов для организации ветвлений и практически не адаптируется к условиям задачи с целью сокращения перебора.

В работе [3] для решения задачи используется полностью целочисленный прямой алгоритм [22] с модифицированным правилом отсечений.

Применение такого алгоритма [3] затруднительно ввиду необходимости решения значительного числа задач линейного программирования общего вида, имеющих большую размерность. Общим недостатком алгоритмов, описанных в [2, 3], является большая размерность вектора двоичных переменных, причем отдельному размещению могут соответствовать $O(4^{|J|^2})$ различных векторов двоичных переменных, что делает эффективность данных алгоритмов особенно низкой для задач, имеющих достаточно много допустимых решений.

В данной работе рассматривается способ [8, 12] декомпозиции задачи размещения прямоугольных объектов на непрерывную и комбинаторную подзадачи. Описанный способ декомпозиции используется для получения необходимых и достаточных условий локального минимума [8] относительно топологии евклидова пространства $\mathbf{R}^{2|J|}$ и построения алгоритма локальной оптимизации в этом пространстве. Построенный алгоритм локальной оптимизации предлагается использовать в генераторе конкурентоспособных вариантов решения задачи размещения прямоугольных объектов при отсутствии ограничений на максимально допустимые расстояния [14]. Для точного решения задачи (1)–(3) предлагается алгоритм по методу ветвей и границ, использующий оригинальные для задач данного класса дерево решений, стратегию ветвления и способ вычисления нижних оценок [12].

1. Декомпозиция задачи размещения прямоугольных объектов

Рассматриваемая задача является дискретно-непрерывной, поэтому представляют интерес различные способы ее декомпозиции на непрерывную и дискретную подзадачи. В данной работе для декомпозиции используется основное свойство допустимого размещения двух прямоугольников $l, k \in J$, определяемое неравенством (2): *проекции прямоугольников на одну из координатных осей должны быть непересекающимися.* В соответствии с этим принципом множество $J \times J$ можно разбить на два симметричных непересекающихся подмножества $\widehat{R}_1$, $\widehat{R}_2$, отнеся к $\widehat{R}_1$ пары объектов, для которых учтено требование непересечения проекций на первую из координатных осей, а к $\widehat{R}_2$ — пары объектов, для которых учтено требование непересечения проекций на вторую из координатных осей.

При фиксации пары отношений $\widehat{R}_1$, $\widehat{R}_2$ задача (1)–(4) распадается на две независимые подзадачи, но их допустимые множества оказываются многосвязными из-за возможности различных упорядочений вдоль каждой оси координат. Для получения частной задачи с выпуклым допустимым множеством достаточно сузить бинарные отношения $\widehat{R}_1$, $\widehat{R}_2$

до максимальных по включению антирефлексивных антисимметричных бинарных отношений R_1, R_2 с взаимно дополняющими симметрическими замыканиями $\overline{(R_1)}$, $\overline{(R_2)}$.

Итак, зафиксируем два антирефлексивных антисимметричных бинарных отношения R_i, $i \in \{1,2\}$, имеющих взаимно дополняющие симметрические замыкания $\overline{(R_i)}$, $i \in \{1,2\}$. Положим

$$\mathbb{F}(R_1, R_2) = \min_{\psi_1 \in \Psi(R_1),\ \psi_2 \in \Psi(R_2)} F(\psi_1, \psi_2), \tag{5}$$

где

$$\Psi(R_i) = \{\psi_i :\ J \to \mathbf{R} \mid (\forall (l,k) \in R_i)\,(\psi_i(l) - \psi_i(k) \geqslant r_i\{l,k\})$$
$$(\forall l, k \in J)\,[|\psi_i(l) - \psi_i(k)| \leqslant d_i\{l,k\}]\}\,. \tag{6}$$

Рассмотрим задачу

$$\mathbb{F}(R_1, R_2) \to \min_{(R_1,R_2):\Psi(R_1)\times\Psi(R_2)\subset\Psi}. \tag{7}$$

Если $(\widehat{R}_1, \widehat{R}_2)$ — решение задачи (7), то решением задачи (1) является

$$\widehat{\psi} = \arg \min_{(\psi_1,\psi_2)\in\Psi(\widehat{R}_1)\times\Psi(\widehat{R}_2)} F(\psi_1, \psi_2). \tag{8}$$

Соотношения (6)–(8) дают двухуровневую схему решения задачи (1)–(3). На верхнем уровне (задача (7)) определяется оптимальное упорядочение. На нижнем уровне (задача (8)) определяется размещение, оптимальное относительно переданного с верхнего уровня упорядочения элементов.

Полезность введенной декомпозиционной схемы заключается в возможности построения на ее основе серии алгоритмов локальной оптимизации для решения исходной задачи. Рассмотрим способ решения задачи (8) нижнего уровня. Данная задача является задачей выпуклого кусочно-линейного программирования. В силу аддитивной сепарабельности целевой функции и блочности системы ограничений по ψ_i, $i \in \{1,2\}$, задача распадается на две независимые подзадачи для каждого значения $i \in \{1,2\}$:

$$\sum_{(l,k)\in R_i} c_i\{l,k\}\,(\psi_i(l) - \psi_i(k))$$
$$+ \sum_{(l,k)\in R_{3-i}} c_i\{l,k\}\,|\psi_i(l) - \psi_i(k)| \to \min_{\psi_i\in\Psi(R_i)}, \tag{9}$$

которые могут быть записаны как задачи линейного программирования следующим образом:

$$\sum_{l\in J}\Big(\sum_{k:(l,k)\in R_i} c_i\{l,k\} - \sum_{k:(k,l)\in R_i} c_i\{l,k\}\Big)\psi_i(l)$$
$$+ \sum_{(k,l)\in R_{3-i}} c_i\{l,k\} p_{kl} \to \min_{(\psi_i,p)}; \tag{10}$$

$$(\forall (l,k) \in R_i)\,[d_i\{l,k\} \geqslant \psi_i(l) - \psi_i(k) \geqslant r_i\{l,k\}]\,; \tag{11}$$

$$(\forall (l,k) \in R_{3-i})\,[-p_{lk} \leqslant \psi_i(l) - \psi_i(k) \leqslant p_{lk}, \\ - d_i\{l,k\} \leqslant \psi_i(l) - \psi_i(k) \leqslant d_i\{l,k\}]. \tag{12}$$

Задачи, двойственные задачам (10)–(12), имеют вид

$$\sum_{(l,k)\in R_i} (r_i\{l,k\}u_i(l,k) - d_i\{l,k\}u_i(k,l)) \\ - \sum_{(l,k)\in R_{3-i}} d_i\{l,k\}\,(u_i(l,k) + u_i(k,l)) \to \max_u \tag{13}$$

при ограничениях

$$(\forall l \in J)\Big(\sum_{k:(l,k)\in R_i} c_i\{l,k\} - \sum_{k:(k,l)\in R_i} c_i\{l,k\} \\ = \sum_{k:(l,k)\in R_i\cup R_{3-i}} (u_i(l,k) - u_i(k,l)) - \sum_{k:(k,l)\in R_i\cup R_{3-i}} (u_i(k,l) - u_i(l,k)) \\ + \sum_{k:\ (l,k)\in R_{3-i}} (w_i(l,k) - w_i(k,l)) - \sum_{k:(k,l)\in R_{3-i}} (w_i(k,l) - w_i(l,k))\Big), \tag{14}$$

$$(\forall (k,l) \in R_{3-i})\,[w_i(k,l) + w_i(l,k) = c_i\{l,k\}, w_i(k,l), w_i(l,k) \geqslant 0]\,, \tag{15}$$

$$(\forall k,l \in J)\,[u_i(k,l) \geqslant 0]\,. \tag{16}$$

Сделав замену переменных

$$(\forall (l,k) \in R_{3-i})\,(2w_i(l,k) = v(l,k) - v(k,l), \\ 2w_i(k,l) = 2c\{k,l\} - v(l,k) + v(k,l))\,,$$

сведем задачи к виду

$$L_i(R_i, R_{3-i}, u_i) = \sum_{(l,k)\in R_i} (r_i\{l,k\}u_i(l,k) - d_i\{l,k\}u_i(k,l)) \\ - \sum_{(l,k)\in R_{3-i}} d_i\{l,k\}\,(u_i(l,k) - u_i(k,l)) \to \max_{u_i\in \mathbb{D}_i}, \tag{17}$$

где $\mathbb{D}_i = \mathbb{D}(R_i, R_{3-i})$ определяется системой ограничений

$$(\forall l \in J)\Big(\sum_{k:(k,l)\in R_i\cup R_{3-i}} c_i\{l,k\} - \sum_{k:(l,k)\in R_i\cup R_{3-i}} c_i\{l,k\} \\ = \sum_{k:k\neq l} (u_i(k,l) - u_i(l,k)) + \sum_{k:(k,l)\notin\overline{(R_i)}} (v_i(k,l) - v_i(l,k))\Big), \tag{18}$$

$$(\forall (k,l) \in R_{3-i})(2c_i\{l,k\} \geqslant v_i(k,l), v_i(l,k) \geqslant 0), \\ (\forall k,l \in J)(u_i(k,l) \geqslant 0).$$

Полученные задачи (17) являются задачами построения оптимального потока в конечной сети, вершины которой — все размещаемые объекты. Пусть (u_i^*, v_i^*) — оптимальные базисные решения, x_i^* — оптимальные векторы двойственных переменных, соответствующих ограничениям — равенствам задач (17), которые в дальнейшем будем обозначать через

$$\Theta_i = \Theta(R_i, R_{3-i}, L_i, u_i^*, x_i^*), \quad i \in \{1, 2\}.$$

Связь между решениями $(\widehat{\psi}_1, \widehat{\psi}_2)$ задачи (8) нижнего уровня и задач Θ_i, $i \in \{1, 2\}$, устанавливает

Теорема 1. $(\forall l \in J)\Big(\widehat{\psi}_1(l) = x_1^*(l),\ \widehat{\psi}_2(l) = x_2^*(l)\Big)$.

Доказательство. В задаче $\Theta(R_i, R_{3-i}, L_i, u_i^*, x_i^*)$, $i \in \{1, 2\}$, условия дополняющей нежесткости для всех $(l, k) \in R_i$ имеют вид

$$\begin{aligned} u_i^*(l,k) = 0 &\Leftrightarrow x_i^*(l) - x_i^*(k) \geqslant r_i\{l,k\}; \\ u_i^*(l,k) \geqslant 0 &\Leftrightarrow x_i^*(l) - x_i^*(k) = r_i\{l,k\}; \\ u_i^*(k,l) = 0 &\Leftrightarrow x_i^*(l) - x_i^*(k) \leqslant d_i\{l,k\}; \\ u_i^*(k,l) \geqslant 0 &\Leftrightarrow x_i^*(l) - x_i^*(k) = d_i\{l,k\}; \end{aligned}$$

а для всех $(l, k) \notin R_i$ —

$$\begin{aligned} u_i^*(l,k) = 0 &\Leftrightarrow |x_i^*(l) - x_i^*(k)| \leqslant d_i\{l,k\}; \\ u_i^*(l,k) > 0 &\Leftrightarrow x_i^*(l) - x_i^*(k) = d_i\{l,k\}; \\ 2c_i\{l,k\} > v_i^*(l,k) > 0 &\Leftrightarrow x_i^*(l) = x_i^*(k); \\ v_i^*(l,k) = 2c_i\{l,k\} &\Leftrightarrow x_i^*(l) \leqslant x_i^*(k); \\ v_i^*(l,k) = 0 &\Leftrightarrow x_i^*(l) \geqslant x_i^*(k). \end{aligned}$$

Таким образом, $\psi = (x_1^*, x_2^*)$ является допустимым решением задачи (9). Из способа построения задач Θ_i, $i \in \{1, 2\}$, следует его оптимальность. Теорема 1 доказана.

Итак, решения задач (8) нижнего уровня непосредственно получаются при решении задач Θ_i, $i \in \{1, 2\}$, построения потока минимальной стоимости в конечной сети N_i. Из условий задачи Θ_i видно, что сеть N_i является мультиграфом с множеством J его вершин и функциями стоимости и пропускной способности дуг, определяемыми по исходной задаче размещения.

Множество дуг сети N_i и функции на них определены следующим образом. Если $(l, k) \in R_i$, то вершинам $l, k \in J$ сети инцидентны две дуги:

1) дуга $(l, k)_1$ с бесконечной пропускной способностью и стоимостью $c_i\{l, k\}$;

2) дуга $(k,l)_2$ с бесконечной пропускной способностью и стоимостью $(-d_i\{l,k\})$.

Если же $(l,k) \in R_{3-i}$, то вершинам $l,k \in J$ сети инцидентны четыре дуги:

1) две дуги $(l,k)_1$ и $(k,l)_1$ с пропускной способностью равной $2c_i\{l,k\}$ и нулевой стоимостью;

2) две дуги $(l,k)_2$ и $(k,l)_2$ с бесконечной пропускной способностью и стоимостью $(-d_i\{l,k\})$.

Отсутствие в исходной задаче ограничений на максимально допустимые расстояния между объектами $l,k \in J$ эквивалентно отсутствию в сетях N_i, $i \in \{1,2\}$, дуг $(l,k)_2$ и $(k,l)_2$. Введение для пары R_{3-i}, $i \in \{1,2\}$, упорядоченности $\psi_i(l) \leqslant \psi_i(k)$ эквивалентно удалению дуг $(k,l)_1$, $(k,l)_2$ из сети N_i.

2. Локальная оптимизация в евклидовом пространстве

В данном разделе рассматриваются вопросы построения локальных экстремумов задачи (1)–(4) относительно топологии евклидова пространства $\mathbf{R}^{2|J|} \ni \psi_1(J) \times \psi_2(J)$. Сущность используемого метода заключается в следующем. Для заданного допустимого решения (ψ_1, ψ_2) строится пара таких бинарных антирефлексивных антисимметричных отношений (R_1, R_2) с взаимно дополняющими симметричными замыканиями, что $\psi_i \in \Psi(R_i)$, $i \in \{1,2\}$. Такими могут быть, например, следующие отношения:

$$(\forall i \in \{1,2\})(R_i = \{(l,k) : \psi_i(l) - \psi_i(k) - r_i\{l,k\} \\ \geqslant |\psi_{3-i}(l) - \psi_{3-i}(k)| - r_{3-i}\{l,k\},\ l,k \in J\}).$$

Построением для некоторого приближенного решения пары отношений (R_1, R_2) и последующим решением задачи (8) можно улучшить приближение к локальному минимуму задачи (1)–(4). Размещение $(\psi_1^*, \psi*_2)$ будет локальным минимумом задачи (1)–(4), если оно является оптимальным решением задачи (8) для всех пар отношений, допускающих данное размещение. Указанный метод построения локального минимума в евклидовом пространстве недостаточно эффективен, так как требуется решение более $O(2^{|J|})$ задач (8). Рассмотрим способ более эффективного алгоритма решения задачи локальной оптимизации.

Теорема 2. *Если задачи Θ_i, $i \in \{1,2\}$, имеют единственное оптимальное решение $(u(R_i, R_{3-i}), w(R_i, R_{3-i}))$ и (ψ_1^*, ψ_2^*) является локальным минимумом задачи (1)–(4), то при $i \in \{1,2\}$*

$$\left(\forall (l,k) \in R_i \setminus R_i^0\right) \left(u(R_i, R_{3-i})(l,k) > 0 \Rightarrow \left|\psi_{3-i}^*(l) - \psi_{3-i}^*(k)\right| < r_{3-i}\right). \tag{19}$$

Доказательство. Допустим противное. Пусть для некоторого $i \in \{1,2\}$ нашлась такая пара объектов $(l,k) \in R_i \setminus R_i^0$, что

$$u(R_i, R_{3-i})(l,k) > 0 \quad \text{и} \quad \psi_{3-i}^*(l) - \psi_{3-i}^*(k) \geqslant r_{3-i}.$$

Построим пару отношений

$$R_i^1 = R_1 \setminus \{(l,k)\}, \quad R_{3-i}^1 = R_2 \cup \{(l,k)\}.$$

Из принятого предположения следует, что

$$\psi_i^* \in \Psi(R_i^1), \quad \psi_{3-i}^* \in \Psi(R_{3-i}^1).$$

Кроме того, очевидно, что

$$(\forall m,n \in J)\left((u(R_{3-i},R_i), w(R_{3-i},R_i)) = (u(R_{3-i}^1,R_i^1), w(R_{3-i}^1,R_i^1))\right).$$

Поэтому

$$L_{3-i}(R_{3-i}, R_i, u(R_{3-i},R_i)) = L_{3-i}(R_{3-i}^1, R_i^1, u(R_{3-i}^1,R_i^1)).$$

Из соотношений

$$(\forall u_i \in \mathbb{D}_i)\ \left(L_i(R_i,R_{3-i},u_i) - L_i(R_i^1,R_{3-i}^1,u_i) = r_i\{l,k\}u_i(l,k) > 0\right),$$
$$\mathbb{D}(R_i,R_{3-i}) \supset \mathbb{D}(R_i^1,R_{3-i}^1)$$

и единственном решении задачи Θ_i следует

$$L_i(R_i,R_{3-i},u_i) > L_i(R_i^1,R_{3-i}^1,u_i).$$

Таким образом, принятое предположение находится в противоречии с оптимальностью решения (ψ_1^*,ψ_2^*).

Противоречивость предположения $\psi_{3-i}^*(l) - \psi_{3-i}^*(k) \geqslant r_{3-i}$ доказывается аналогично с тем отличием, что полагается

$$R_i^1 = R_1 \setminus \{(l,k)\}, \quad R_{3-i}^1 = R_2 \cup \{(l,k)\}.$$

Теорема 2 доказана.

Теорема 3. *Если размещение* (ψ_1^*,ψ_2^*) *и оптимальные решения* (u_i,w_i) *задач* Θ_i, $i \in \{1,2\}$, *удовлетворяют условию* (19), *то* (ψ_1^*,ψ_2^*) *является локальным минимумом задачи* (1)–(4).

Доказательство. Из (19) следует, что для $i \in \{1,2\}$ ненулевые оптимальные потоки имеются только между объектами

$$(l,k):\ (l,k) \in R_i \setminus R_i^0 \Rightarrow \left|\psi_{3-i}^*(l) - \psi_{3-i}^*(k)\right| < r_{3-i},$$

любое же изменение отношения между указанными парами объектов делает данное размещение недопустимым. Теорема 3 доказана.

Данная теорема дает достаточные условия локального экстремума задачи (1)–(4). Конструктивность этих условий устанавливает следующая

Теорема 4. *Если размещение* (ψ_1^*, ψ_2^*) *является локальным минимумом задачи* (1)–(4), *то существует пара бинарных отношений* (R_1, R_2) *такая, что* $\psi_i^* \in \Psi(R_i)$, $i \in \{1,2\}$, *и оптимальные решения* (u_i, w_i) *задач* Θ_i, $i \in \{1,2\}$, *удовлетворяют условию* (19).

Доказательство. Приведем конструктивное доказательство данной теоремы. Изобразим алгоритм **EuLocAlg** построения пары бинарных отношений, существование которой утверждается в теореме.

Алгоритм EuLocAlg
begin
$R_1 := \{(l,k) : \psi_1(l) - \psi_1(k) - r_1\{l,k\} \geqslant |\psi_2(l) - \psi_2(k)| - r_2\{l,k\}, l,k \in J\}$;
$R_2 := \{(l,k) : \psi_2(l) - \psi_2(k) - r_2\{l,k\} \geqslant |\psi_1(l) - \psi_1(k)| - r_1\{l,k\}, l,k \in J\}$;
$\Theta(R_1, R_2, L_1, u_1, x_1)$; $\psi_1 := x_1$; $\Theta(R_2, R_1, L_2, u_2, x_2)$; $\psi_2 := x_2$;
$i := 1$; $Ind := 0$;
repeat
  $\mathrm{Inc}(Ind)$; $i := 3 - i$;
  for $(l,k) \in R_i \setminus R_i^0 :\ u_i(l,k) > 0$ **do**
    if $(\psi_{3-i}(l) - \psi_{3-i}(k) \geqslant r_{3-i}(l,k))$ **then begin**
      $R_i = R_i \setminus \{(l,k)\}$; $R_{3-i} = R_{3-i} \cup \{(l,k)\}$; $\Theta(R_i, R_{3-i}, u_i, x_i)$;
      if $(\psi_i <> x_i)$ **then begin** $\psi_i := x_i$; $Ind := 0$; **break**; **end**;
      end
    else if $(\psi_{3-i}(k) - \psi_{3-i}(l) \geqslant r_{3-i}(l,k))$**then begin**
      $R_i = R_i \setminus \{(l,k)\}$; $R_{3-i} = R_{3-i} \cup \{(k,l)\}$; $\Theta(R_i, R_{3-i}, u_i, x_i)$;
      if $(\psi_i <> x_i)$**then begin** $\psi_i := x_i$; $Ind := 0$; **break**; **end**;
      end;
  until $(Ind = 2)$;
stop
end

Если первоначальное решение не является локальным минимумом задачи (1)–(4), то приведенный алгоритм одновременно строит и соответствующее локально-оптимальное решение. Основной цикл **repeat** алгоритма **EuLocAlg** может быть завершен только в случаях, когда при текущих отношениях R_1, R_2 для пары объектов (l,k) таких, что $(l,k) \in R_i \setminus R_i^0$, $\psi_{3-i}(l) - \psi_{3-i}(k) = r_{3-i}(l,k)$, будет выполнено достаточное условие оптимальности (19): $u_i(l,k) = 0$.

Докажем конечность данного алгоритма. Число операций в цикле **for** не превосходит величины $O(p)$, где p — трудоемкость решения задачи об оптимальном потоке. Так как число активных дуг в базисных решениях задач Θ_i не больше $|J|$, то число повторений цикла **for** при однократном выполнении тела цикла **repeat** не превосходит величины $O(|J|)$. Таким образом, сложность цикла **repeat** не превосходит величины $O(|J|p)$.

При двукратном выполнении тела цикла **repeat** без прерываний внутреннего цикла **for** алгоритм завершает работу. Прерывание выполнения тела цикла **for** возможно только при уменьшении целевого функционала соответствующей задачи об оптимальном потоке, а следовательно, и изменении текущего варианта размещения.

Целевой функционал этих задач ограничен сверху величиной $M^2|J|^2$, где M — максимальное из чисел в исходных данных задачи. Из целочисленности оптимальных базисных решений задачи об оптимальном потоке и неотрицательности целевого функционала следует, что число прерываний цикла **for** не превосходит величины $M^2|J|^2$. Следовательно, общее число операций в алгоритме не превосходит $O(M^2|J|^3p)$.

Так как задача об оптимальном потоке разрешима за псевдополиномиальное время, то алгоритм **EuLocAlg** строит решения задачи (1)–(4), локально-оптимальные относительно евклидовой топологии в пространстве решений, за псевдополиномиальное время. Теорема 4 доказана.

В доказательстве теоремы 4 также построен и обоснован псевдополиномиальный алгоритм локальной оптимизации при евклидовой топологии пространства решений задачи (1)–(4). Таким образом, доказана

Теорема 5. *Алгоритм* **EuLocAlg** *корректно решает проблему локальной оптимизации для задачи* (1)–(4) *относительно евклидовой топологии пространства решений. Его сложность не превосходит* $O(M^2|J|^3p)$.

Итак, построение локально-оптимальных относительно топологии евклидова пространства размещений сведено к решению последовательности задач построения оптимального потока. При этом условия двух последовательных задач отличаются лишь параметрами одной из дуг сети и параметрами вершин, инцидентных этой дуге. В работе [9] указан способ построения по оптимальному базисному решению первой из рассматриваемых задач базисного решения второй задачи. Это позволяет использовать алгоритмы решения сетевых транспортных задач, построенные по схеме метода потенциалов. Опыт решения сетевых транспортных задач показывает, что наиболее эффективными алгоритмами их решения являются именно алгоритмы, построенные по схеме метода потенциалов. Эффективность решения задач, используемых алгоритмом **EuLocAlg**, можно сделать еще более высокой, если применить упорядочение информации о дугах сети, описанное, например, в [9, 13].

3. Генератор конкурентоспособных вариантов решения задачи размещения прямоугольных объектов

В практических задачах длина связывающей сети является определяющим критерием качества размещения, но в общем случае не единственным. Возможным способом решения этих задач является генерация конкурентоспособных вариантов и их последующая оценка в соответствии с комплексным критерием качества и системой предпочтений лица, принимающего решение. Генерация конкурентоспособных вариантов возможна с помощью использования методов случайного поиска и алгоритма **EuLocAlg**. Если ограничения задачи (1)–(4) достаточно жесткие, то вероятность получения допустимого решения будет низкой. В этом случае более целесообразным оказывается поиск подходящих решений с помощью метода последовательно-одиночного упорядочения — аналога метода последовательно-одиночного размещения.

Основным отличием метода последовательно-одиночного упорядочения от метода последовательно-одиночного размещения является жесткая фиксация не координат ранее размещенных объектов, а их упорядочения, что приводит к построению размещений более высокого качества. Эффективная реализация метода последовательно-одиночного упорядочения позволяет использовать его для генерации вариантов решения задачи (1)–(4) по схемам, аналогичным схемам использования метода последовательно-одиночного размещения.

Для решения задачи (1)–(2) можно использовать случайный поиск в пространстве $\mathbf{R}^{2|J|} \ni \psi_1(J) \times \psi_2(J)$ в сочетании с алгоритмом **EuLocAlg**. Основной целью адаптивных алгоритмов [17] является не только генерация вариантов размещении, но и поиск оптимального решения. Рассмотрим алгоритм **RandLocAlg**, сочетающий использование случайного поиска и эвристики, предназначенный для быстрой генерации конкурентоспособных вариантов размещения (т. е. близких по функционалу к оптимальному решению).

Размещение ψ, построенное алгоритмом **RandLocAlg**, является допустимым и локально-оптимальным относительно топологии евклидова пространства $\mathbf{R}^{2|J|} \ni \psi_1(J) \times \psi_2(J)$ вариантом решения задачи (1)–(2). Результативность и конечность алгоритма **RandLocAlg** очевидны. Применяемая эвристика состоит в использовании так называемой механической аналогии [18, 27]. Размещаемые объекты интерпретируются материальными точками. Между всеми парами $l, k \in J$ вводятся силы притяжения и силы отталкивания. Проекции силы притяжения между парой точек (l, k) на координатные оси пропорциональны проекциям расстояния и стоимостям коммуникаций между этими точками, а силы отталкивания между ними пропорциональны величине невязки в соответствующем ограничении (2).

В цикле по переменной j фактически моделируются движение такой связной системы материальных точек. При этом на первых итерациях из-за малости величины ε скорость взаимного удаления слабо связанных объектов будет выше скорости взаимного удалении сильно связанных, что обеспечивает относительную близость координат сильно связанных объектов в окончательном решении.

Алгоритм RandLocAlg (Output: $\psi : J^2 \to \mathbf{R}^2$)
begin
$\alpha := \frac{1}{|J|^2} \sum_{i \in \{1,2\}} \sum_{l,k \in J} c_i(l,k)$; $\gamma := \frac{1}{|J|^2} \sum_{i \in \{1,2\}} \sum_{l,k \in J} r_i(l,k)$; $\epsilon := 0.1\gamma$;
RandDraft(ε, ψ); {Получить случайный ε-наброс}
for $j := 1$ **upto** $\log_2(|J|)$ **do begin**
  for $i \in \{1,2\}$ **do for** $l \in J$ **do begin**
   $\psi_i^1(l) := 0$;
   for $k \in J$ **do begin**
    $d := \min \begin{bmatrix} r_i\{l,k\} - |\psi_i^1(l) - \psi_i^1(k)| \\ r_{3-i}\{l,k\} - |\psi_{3-i}^1(l) - \psi_{3-i}^1(k)| \end{bmatrix}$;
    if $(d < 0)$ **then** $d := c_i\{l,k\} + d \cdot \frac{\alpha}{r_{3-i}\{l,k\}}$ **else** $d := c_i\{l,k\}$;
    if $\psi_i(l) > \psi_i(k)$ **then** $\mathrm{Inc}(\psi_i^1(l), d)$ **else** $\mathrm{Dec}(\psi_i^1(l), d)$;
    end;
   end:
  $\psi := \psi - \frac{\gamma \cdot \psi^1}{j \cdot ||\psi^1||}$;
  end;
EuLocAlg (ψ);
stop
end

4. Применение схемы метода ветвей и границ для размещения прямоугольных объектов

Алгоритмы решения задачи размещения прямоугольных объектов, основанные на случайном поиске и эвристиках, применимы в случаях, когда допустимая область имеет достаточно большое число локальных экстремумов. В противном случае данные методы просто не находят допустимых решений. Следовательно, для решения задач такого типа необходим алгоритм, который за конечное число шагов находил бы допустимый, а затем и оптимальный или близкий к нему по стоимости вариант, или же устанавливал неразрешимость задачи.

Универсальным методом решения комбинаторных задач является метод ветвей и границ [5–7]. Хотя алгоритмы ветвей и границ обычно являются более эффективными, чем перебор, все же их требования в вычислительных ресурсах растут как экспоненты или полиномы высокой степени от размера задачи. В работе [25] доказана $\mathbb{NP}$-полнота одномерной задачи размещения, являющейся частным случаем задачи размещения прямоугольных объектов. Это дает основание считать, что потребности в вычислительных ресурсах для нахождения точных решений большого числа таких задач не могут быть ограничены полиномом от размерности задачи. В такой ситуации реальные задачи не могут быть решены точно. Основой для построения конечных алгоритмов могут быть различные комбинаторные и целочисленные объекты, в частности:

1) векторы двоичных переменных, введенные в [2, 3];

2) пары отношений (R_1, R_2) на множестве $|J|$, введенные выше.

Основным недостатком использования отмеченных выше объектов является экспоненциальный рост их числа для одного заданного размещения, что делает алгоритмы, использующие данные структуры для построения дерева поиска, практически неприменимыми для решения реальных задач.

В данной работе для решения задачи (1)–(4) предлагается релаксационный подход. Первоначально задача решается для множества $\mathbb{D}_0^0$, определяемого системой ограничений (3)–(4). В случае, если найденное решение удовлетворяет условиям (2), оно является решением задачи (1)–(4). В противном случае выбирается пара объектов (l, k), для которых не выполнено ограничение (2), и производится элиминация множества

$$\left\{(\psi_1, \psi_2):\ (\forall l, k \in J)\Big(\max_{i\in\{1,2\}}[|\psi_i(l) - \psi_i(k)| - r_i\{l,k\}] < 0\Big)\right\}$$

из множества $\mathbb{D}_0^0$ с последующим разбиением полученного множества на выпуклые непересекающиеся подмножества $\mathbb{D}_1^n$, где $n = 1, 2, 3, 4$. Далее описанный процесс повторяется для каждого полученного подмножества. Лучшее из построенных таким образом допустимых решений задачи (1)–(4) является ее оптимальным решением.

Сократить трудоемкость организованного таким образом поиска позволяет метод ветвей и границ, для описания которого достаточно определить дерево поиска и функцию нижней оценки. *Дерево поиска* имеет следующий вид. Каждой n-й вершине m-го уровня ставится во взаимно-однозначное соответствие множество допустимых решений $\mathbb{D}_m^n$. Вершине дерева решений соответствует множество

$$\mathbb{D}_0^0 = \mathbb{D}(R_1^0, R_2^0) \times \mathbb{D}(R_2^0, R_1^0).$$

В случае, если $\mathbb{D}_m^n \neq \varnothing$ и

$$\psi^{mn} = \arg\min_{\psi\in\mathbb{D}_m^n} \mathbb{F}(\psi) \notin \Psi, \tag{20}$$

вершина $\mathbb{D}_m^n$ имеет четыре потомка

$$\mathbb{D}_{m+1}^{4n+2|i-i^*|+|j-j^*|} = \mathbb{D}_m^n \bigcap \Big\{\psi : \ s_{ij}^{mn}(\psi_i(l) - \psi_i(k) - r_i\{l,k\}) \geqslant 0,$$
$$|i-i^*|(|\psi_{3-i}(l) - \psi_{3-i}(k)| - r_{3-i}(l,k) + 1) \leqslant 0\Big\}, \quad i,j \in \{1,2\}, \tag{21}$$

где

$$i^* = \arg\max_{i\in\{1,2\}} \{|\psi_i^{mn}(l) - \psi_i^{mn}(k)| - r_i\{l,k\}\}; \tag{22}$$

$$|\psi_{i^*}^{mn}(l) - \psi_{i^*}^{mn}(k)| - r_{i^*}(l,k) < 0; \tag{23}$$

$$j^* = \frac{1}{2}(\operatorname{sign}(\psi_{i^*}^{mn}(l) - \psi_{i^*}^{mn}(k)) + 3); \tag{24}$$

$$s_{ij}^{mn} = (-1)^{j+1}\operatorname{sign}(\psi_{i^*}^{mn}(l) - \psi_{i^*}^{mn}(k)), \quad i,j \in \{1,2\}; \tag{25}$$

$$\operatorname{sign}(a) = \begin{cases} 1, & \text{если } a > 0, \\ -1, & \text{если } a \leqslant 0. \end{cases}$$

Легко проверить, что имеет место равенство

$$\bigcup_{p,q\in\{0,1\}} \mathbb{D}_{m+1}^{4n+2p+q} = \mathbb{D}_m^n \setminus \Big\{\max_{i\in\{1,2\}}[|\psi_i(l) - \psi_i(k)| - r_i\{l,k\} + 1] \leqslant 0\Big\}, \tag{26}$$

при этом для всех $p_0, q_0, p_1, q_1 \in \{0,1\} : \ (p_0,q_0) \neq (p_1,q_1)$

$$\mathbb{D}_{m+1}^{4n+2p_0+q_0} \cap \mathbb{D}_{m+1}^{4n+2p_1+q_1} = \varnothing. \tag{27}$$

В качестве нижней оценки для вершины $\mathbb{D}_m^n$ примем величину

$$L(\mathbb{D}_m^n) = \min_{\psi\in\mathbb{D}_m^n} \mathbb{F}(\psi).$$

Корректность данного определения нижней оценки также очевидна. Для проверки условия (20) и определения i^* и j^*, удовлетворяющих условиям (22)–(24), можно использовать различные методы. Наиболее простым представляется упорядочение множества всех пар $\{l,k\} \subset J$. В этом случае на m-м уровне следует производить ветвление по m-й паре объектов, даже если для них оказывается нарушенным условие (23), что может привести к увеличению числа ветвей дерева решений. Способ упорядочения пар объектов $\{l,k\} \subset J$ существенно влияет на эффективность работы алгоритма. Эксперименты показали целесообразность упорядочения по величине $c_1(l,k)r_1\{l,k\} + c_2(l,k)r_2\{l,k\}$.

Вершина $\mathbb{D}_{m+1}^{4n}$ имеет не большее значение функции нижней оценки, чем остальные потомки вершины $\mathbb{D}_m^n$. Системы ограничений множеств $\mathbb{D}_{m+1}^{4n}$ и $\mathbb{D}_{m+1}^{4n+2}$ отличаются от системы ограничений множества $\mathbb{D}_m^n$ одним ограничением, а системы ограничений множеств $\mathbb{D}_{m+1}^{4n+1}$ и $\mathbb{D}_{m+1}^{4n+3}$ — двумя.

Согласно изложенному выше наиболее целесообразной стратегией обхода дерева поиска, особенно на этапе получения первого допустимого решения и верхней оценки, является обход в глубину. Алгоритм **BBLocAlg** позволяет находить такое допустимое решение L, если оно существует, что $(L - L^*)/L^* < \varepsilon$, где L^* — оптимальное решение. В алгоритме используется стратегия обхода дерева поиска в глубину. Входными данными являются список $\mathbb{L}$ пар $\{l, k\} \subset J$, определяющий порядок ветвления, и переменная ε, задающая относительную погрешность вычисления минимума функционала.

Алгоритм BBLocAlg

Input: $\mathbb{L} : \mathbf{N} \to \{\{l,k\} : l,k \in J\}$ {порядок ветвления};
$\varepsilon \in [0,1]$ {точность};

Output: $\psi^* : J \to \mathbf{R}^2$ {оптимальный вариант размещения}

Variables: $u \in \mathbf{R}$; $M = \dfrac{|J|(|J|-1)}{2}$;

Procedure BranchLoc($m \in \mathbf{N}$, $\mathbb{D} \subset \Psi$) **begin**

$\psi := \arg\min\limits_{\psi \in \mathbb{D}} \mathbb{F}(\psi)$; $L := \min\limits_{\psi \in \mathbb{D}} \mathbb{F}(\psi)$; **if** $(L(1+\epsilon) > u)$ **then exit**;

if $(m = M)$ **then begin** $u := L$; $\psi^* := \psi$; **exit**; **end**;

$m := m + 1$; $\{l,k\} := \mathbb{L}(m)$; $s := 1$;

$n := \arg\max\limits_{i \in \{1,2\}} [|\psi_i(l) - \psi_i(k)| - r_i\{l,k\}]$;

for $i \in \{1,2\}$ **do begin**

for $j \in \{1,2\}$ **do begin**

$\mathbb{D}1 := \mathbb{D} \cap \{\psi : s[\psi_n(l) - \psi_n(k)] \geqslant r_n\{l,k\}\}$;

BranchLoc(m, $\mathbb{D}1$); $s := -s$;

end;

$\mathbb{D} := \mathbb{D} \cap \{\psi : s|\psi_n(l) - \psi_n(k)| \leqslant r_n\{l,k\} - 1\}$; $n := 3 - n$;

end;

end;

begin

$m := 0$; $u := +\infty$; $\mathbb{D} := \{\psi : J \to \mathbf{R}^2\}$;

BranchLoc(m, $\mathbb{D}$); **stop**;

end

Основу алгоритма **BBLocAlg** составляет рекурсивная процедура BranchLoc. Параметрами процедуры являются переменная m, определяющая дугу из списка $\mathbb{L}$, по которой производится ветвление, и объект $\mathbb{D}$, определяющий задачу нижнего уровня. В процедуре BranchLoc вычисляется нижняя оценка L текущего решения. Если текущее решение оказывается лучшим из допустимых решений, то производится уточнение верхней оценки u, а текущее решение принимается за рекордное. В противном случае, если текущее решение является перспективным (т. е. $L(1+\varepsilon) < u$), то последовательно решаются 4 задачи следующего уровня. Легко заметить, что последовательно решаемые задачи отличаются не более чем двумя ограничениями.

Результативность и конечность алгоритма **BBLocAlg** очевидны. Очевидно также, что возможна реализация, которая имеет пространственную сложность не более $O(|J|^2)$. Верхней оценкой временной сложности алгоритма является величина $O((|J|!)^2)$, что ниже оценки известных точных алгоритмов [1, 2], равной $O(4^{|J|^2})$.

Заключение

Предложенный в статье способ иерархической декомпозиции задачи размещения прямоугольных объектов с минимальной длиной связывающей их сети на задачу оптимального упорядочения (верхний уровень) и две задачи построения оптимального потока (нижний уровень) позволяет строить множество методов решения и анализа исходной задачи. В частности, предложены:

1) алгоритм **EuLocAlg** построения локальных экстремумов;

2) алгоритм **RandLocAlg**, основанный на использовании случайного поиска и эвристики;

3) алгоритм **BBLocAlg** по методу ветвей и границ.

Наиболее трудоемким в построенных алгоритмах является решение или установление неразрешимости задач (20). Задача, двойственная (20), распадается на две независимые сетевые транспортные задачи линейного программирования. Так как условия двух последовательно решаемых алгоритмом задач вида (20) отличаются не более чем одним дополнительным ограничением, то оптимальное базисное решение задачи, двойственной одной из этих задач, будет хорошим базисным решением задачи, двойственной другой из этих задач. Методы повышения эффективности решения такой последовательности возмущенных задач рассмотрены в [9, 13].

ЛИТЕРАТУРА

1. **Жак С. В., Зинченко А. Б.** Решение некоторых задач геометрического размещения модифицированным методом ветвей и границ // Математическое обеспечение АСУП: Тез. докл. Второго Всесоюз. семинара. М.: ИПУ, 1975. С. 27–28.
2. **Жак С. В., Зинченко А. Б.** Комбинаторные методы решения задачи размещения помещений в производственном здании // Автоматизация архитектурно-строительного проектирования промышленных предприятий. Ростов на/Д.: РИСИ, 1979. С. 56–57.

3. **Зинченко А. Б.** Об одной задаче оптимального геометрического размещения // Численные методы нелинейного программирования: Тез. докл. III Всесоюз. семинара. Харьков, 1979. С. 15–16.
4. **Зинченко А. Б.** Локальный алгоритм для задачи размещения // Автоматизация архитектурно-строительного проектирования промышленных предприятий. Ростов н/Д.: РИСИ, 1986. С. 136–141.
5. **Коглер В., Штиглиц К.** Перечислительные и итеративные алгоритмы // Теория расписаний и вычислительные машины. М.: Наука, 1984. С. 249–317.
6. **Корбут А. А., Сигал И. Х.** Финкельштейн Ю. Ю. Метод ветвей и границ: (Обзор теории, алгоритмов, программ и приложений) // Math. Operationsforsch Statist. Ser. Optim. 1980. V. 8, N 2. P. 253–280.
7. **Корбут А. А., Финкельштейн Ю. Ю.** Дискретное программирование. М.: Наука, 1969.
8. **Панюков А. В.** Алгоритм локальной оптимизации для задачи размещения прямоугольных объектов с минимальной длиной связывающей их сети // Изв. АН СССР. Техн. кибернетика. 1981. № 6. С. 180–184.
9. **Панюков А. В.** Метод решения возмущенной транспортной задачи на сети // Методы и программы решения оптимизационных задач на графах и сетях. Ч. 2: Теория, алгоритмы: Тез. докл. II Всесоюз. совещ. (Улан-Удэ, 1982). Новосибирск: ВЦ СО АН СССР, 1982. С. 113–114.
10. **Панюков А. В.** Декомпозиция задачи размещения прямоугольных объектов // Декомпозиция и координация в сложных системах: Тез. докл. Всесоюз. конф. Ч. 1. Челябинск: ЧПИ, 1986. С. 99–100.
11. **Панюков А. В.** Алгоритмы размещения прямоугольных объектов // Декомпозиция и координация в сложных системах: Материалы Всесоюз. конф. Челябинск: ЧПИ, 1987. С. 80–97.
12. **Панюков А. В.** Размещение прямоугольных объектов // Декомпозиция и координация в сложных системах: Материалы Всесоюз. конф. Челябинск: ЧПИ, 1987. С. 80–89.
13. **Панюков А. В.** Упорядоченное изучение базисного дерева в прямых алгоритмах для транспортной задачи // Междунар. Сибирская конф. по исследованию операций: Материалы конф. Новосибирск: Изд-во Ин-та математики, 1998. С. 44.
14. **Панюков А. В., Петренко С. А.** Формирование конкурентоспособных вариантов схемы компоновки генерального плана предприятия на ЭВМ // Изв. вузов. Строительство и архитектура. 1987. № 8. С. 119–121.
15. **Плужников Л. Н.** Математическое обеспечение некоторых задач размещения и компоновки // Автоматизация проектирования как комплексная проблема совершенствования проектного дела в стране. М.: ЦДНТП, 1973. С. 201–207.

16. **Плужников Л. Н., Рачинский В. С.** Математическое обеспечение АСП коммуникаций и генерального плана промышленного узла // Автоматизированные системы проектирования. М.: МДЦТП, 1975. С. 185–190.
17. **Растригин Л. А.** Современные принципы управления сложными объектами. М.: Сов. радио, 1980.
18. **Селютин В. А.** Автоматизация проектирования топологии БИС. М.: Радио и связь, 1983.
19. **Стоян Ю. Г., Гиль Н. И.** Методы и алгоритмы размещения плоских геометрических объектов. Киев: Наук. думка, 1977.
20. **Стоян Ю. Г., Соколовский В. З.** Решение некоторых многоэкстремальных задач методом сужающихся окрестностей. Киев: Наук. думка, 1979.
21. **Benders J. F.** Partitioning procedures for solving mixed variables programming problems // Numer. Math. 1962. N. 4. P. 139–151.
22. **Ben-Israel A., Charnes A.** On some problems of Diophantine programming // Cahiers Centre Études Rech. Opér. 1962. V. 4. P. 121–135.
23. **Francis R. L., White J. A.** Facilities layout and location: an analitical approach. Englewood Cliffs. NJ: Prentice-Hall, 1974.
24. **Panyukov A. V.** The study of basis tree for primal transhipment algorithm // Combinatorial Optimization 94: Program & Abstracts of Intern. Conf. Amsterdam, 1994.
25. **Picard J., Queyranne M.** On the one-dimensional allocation problem // Oper. Res. 1981. V. 29, N 2. P. 371–391.
26. **Preas B. T., van Cleemput W. M.** Placement algorithms for arbitrargily shaped bloks // Proc. of the 16th DAC. 1979. P. 474–480.
27. **Quinn N. R., Breuer M. A.** Forced directed component placement procedure for PCB // IEEE Trans. on Circuits and Systems. 1979. V. CAS-26, N 6. P. 377–388.

Адрес автора:

Южно-Уральский
государственный университет,
пр. Ленина, 76,
454080 Челябинск, Россия.
E-mail: panyukov@inf.susu.ac.ru

Статья поступила
26 июня 2000 г.,
переработанный вариант —
22 ноября 2000 г.