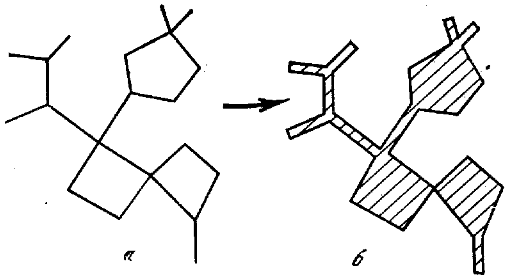

а *б* **Рис. 10**

Из этих точек оставим для рассмотрения только одну (произвольную) точку, а остальные учитывать не будем. Таким образом определено взаимно-однозначное отображение множества $V(\Gamma)$ на выделенные $|V(\Gamma)|$ точек окружности O, которое является искомым. Теорема о расположении на окружности установлена.

А. В. Карзанов

ЭКОНОМНЫЕ РЕАЛИЗАЦИИ АЛГОРИТМОВ ЭДМОНДСА НАХОЖДЕНИЯ ПАРОСОЧЕТАНИЯ МАКСИМАЛЬНОЙ МОЩНОСТИ И МАКСИМАЛЬНОГО ВЕСА

1. Рассматривается граф $G = [V(G),\ U(G)]$ с множеством вершин $V(G)$, $|V(G)| = n$, множеством ребер $U(G)$, $|U(G)| = p$. Множество $P \subset U(G)$ называется паросочетанием, если никакие два ребра из P не имеют общих концов.

Задача А. Найти в G паросочетание M, мощность которого (т. е. $|M|$) максимальна (паросочетание M называется *максимальным*).

Задача Б. Заданы «веса» ребер — функция $c(u) \geqslant 0$, $u \in U(G)$. Найти паросочетание M^c, «вес» которого (т. е. $\sum\limits_{u \in M^c} c(u)$, максимален (паросочетание *максимального веса*).

Эдмондс [1], [2] впервые предложил эффективные алгоритмы решения обеих задач. Анализируя эти алгоритмы, можно установить, что оценка числа действий на решение задачи **А** — $O(n^2p)$, задачи **Б** — $O(n^3p)$, требуемая рабочая память — $O(np)$ «ячеек» (единицей информации и операндом считается многоразрядное «слово», помещающееся в ячейке). Алгоритмы решения задачи **А** приведены также в [3], [4], оценка числа действий для них — $O(n^2p)$, требуемая рабочая память — $O(p)$.

В настоящей работе предлагаются реализации алгоритмов Эдмондса (по существу это модифицированные алгоритмы). Оценка реализации для задачи **А** — $O(n^3)$, для задачи **Б** — $O(n^3 + np\cdot\log_2 n)$, оценка рабочей памяти — $O(n^2)$.

2. Пусть P — паросочетание в графе G. Вершины, не инцидентные ребрам из P, назовем свободными. Ребра, принадлежащие P, и только их назовем выделенными.

Простая цепь (быть может, вырожденная) называется чередующейся, если в любой паре ее соседних ребер ровно одно выделенное. Невырожденная чередующаяся цепь называется увеличивающей (ув.), если ее концевые вершины — свободные.

Критерий максимальности паросочетания дает теорема Бержа [5]: P максимально, если, и только если в G нет ув. чередующейся цепи.

При наличии ув. чередующейся цепи L можно перейти к паросочетанию на единицу большей мощности, поменяв ролями выделенные и невыделенные ребра этой цепи. Такую процедуру назовем *перекраской* вдоль L.

На основании теоремы Бержа максимальное паросочетание можно получить из исходного (быть может, пустого) паросочетания P путем последовательного построения паросочетаний $P_0 = P, P_1, P_2, \ldots, P_m$; каждое паросочетание получается из предыдущего перекраской вдоль некоторой ув. чередующейся цепи, P_m не допускает существования такой цепи.

3. Алгоритм Эдмондса решения задачи А. Пусть уже построено паросочетание P. Выбирается некоторая свободная вершина r и решается задача $\mathbf{A}_r$: найти ув. чередующуюся цепь, одним концом которой является r, либо показать, что такой цепи не существует. В последнем случае будем говорить, что задача $\mathbf{A}_r$ решена отрицательно. Решение задачи $\mathbf{A}_r$ назовем этапом в алгоритме.

Вершина x называется внешней, если существует чередующаяся цепь четной длины [1], соединяющая r и x. Вершина x называется внутренней, если она не внешняя и существует чередующаяся цепь, соединяющая x и r нечетной длины. Обозначим E — множество внешних, I — внутренних вершин. В алгоритме решения задачи $\mathbf{A}_r$ фактически последовательно отыскиваются внешние и внутренние вершины и для каждой такой вершины x указывается способ нахождения чередующейся цепи $L_{x,r}$ (из x в r) четной (нечетной) длины. Как только обнаруживается свободная вершина x, задача $\mathbf{A}_r$ решена ($L_{x,r}$ — искомая ув. чередующаяся цепь).

Пусть множество E и I построены и I не содержит свободной вершины. Тогда задача $\mathbf{A}_r$ решена отрицательно. Из теории Эдмондса следует:

Теорема 1. Если в множестве I нет свободных вершин и L — произвольная ув. чередующаяся цепь, то L не содержит вершин из E и I. ■

Из теоремы 1 легко следует, что нахождение увеличивающих чередующихся цепей на последующем

[1] Длиной цепи считается число дуг в ней.

этапе достаточно проводить на подграфе $G_{\{V(G)\setminus(E\cup I)\}}$ [2], что и делается в алгоритме. Тем самым задача $\mathbf{A}_r$ для каждого $r \in V(G)$ решается не более одного раза, следовательно, число этапов в алгоритме не более n.

Остановимся на задаче $\mathbf{A}_r$. Простой цикл C, содержащий $2k+1$ ребер ($k \geqslant 1$ — целое), k из которых принадлежат P, назовем чередующимся. Ровно одна его вершина не инцидентна ребрам из P, лежащим в C, назовем ее *корнем* цикла C и обозначим $r(C)$. Важной операцией в алгоритме является стягивание чередующегося цикла: замена в G множества его вершин одной новой вершиной (обозначим ее также C), возникающие при этом кратные ребра «склеиваются», петли выбрасываются. Новый граф обозначим G^C. Ребро $(C, x) \in U(G^C)$, возникшее из ребра $(y, x) \in U(G)$, $y \in V(C)$, назовем *производным* (от ребра (y, x)). Зададим отображение $\varphi^C : V(G) \to V(G^C)$, при котором вершина $x \in V(G)$ переходит в себя, если $x \notin V(C)$, и в C, если $x \in V(C)$. Отображение φ^C естественно продолжается до отображения $\overline{\varphi}^C : G \to G^C$. Образы выделенных ребер графа G будем считать выделенными в G^C, их множество обозначим P^C. Можно доказать, что P^C — паросочетание.

Чередующийся цикл C назовем правильным, если существует чередующаяся цепь $L_{r(C),r}$ (из $r(C)$ в r) четной длины, не пересекающаяся с C, кроме как в вершине $r(C)$. Из теории Эдмондса следует:

Теорема 2. Если C — правильный цикл, то $E = (\varphi^C)^{-1}(E^C)$, $I = (\varphi^C)^{-1}(I^C)$, где E^C (I^C) — множество внешних (внутренних) вершин графа G^C. ■

Пусть $x \in V(C)$. Ту из двух цепей, соединяющих в C вершину x с $r(C)$, которая имеет четную длину, обозначим $L^C_{x,r(C)}$. Очевидно, $L_{x,r} = L^C_{x,r(C)} \cup L_{r(C),r}$ — чередующаяся цепь четной длины, следовательно, вершина x — внешняя.

[2] $G_{\{S\}}$ обозначает подграф графа G, натянутый на множество вершин $S \subset V(G)$.

Теорема 2 позволяет свести поиск внешних (внутренних) вершин графа G к нахождению внешних (внутренних) вершин более простого графа, получающегося из G последовательными стягиваниями правильных циклов.

Если $(x, y) \in P$, то будем обозначать $x = p(y)$, $y = p(x)$.

Чередующимся деревом в графе G назовем дерево $T = [V(T), U(T)]$ с корнем r такое, что: а) для каждой вершины $x \in V(T)$ цепь, соединяющая в T вершины x и r, является чередующейся (эту цепь назовем *возвратной* и обозначим $L(x)$); б) возвратная цепь каждой висячей вершины имеет четную длину. Множество вершин в T, возвратные цепи которых имеют четную (нечетную) длину, обозначим $E(T)$ (соответственно $I(T)$). Отметим простые свойства чередующегося дерева T: а) $E(T) \subset E$; б) вершины ветвления дерева (т. е. вершины, инцидентные не менее трем ребрам дерева) принадлежат $E(T)$; в) если $x \in V(T)$ и $x \neq r$, то $y = p(x) \in V(T)$.

Алгоритм решения задачи A_r. Считаем, что поиск ув. чередующихся цепей осуществляется в графе $\overline{G}$, полученном из G выкидыванием внешних и внутренних вершин, найденных при отрицательном решении задач на предыдущих этапах. Вначале полагаем $T = \{r\}$. Пусть уже несколько итераций проведено и построены графы $G_0 = \overline{G}$, $G_1 = G_0^{C_1}$, $G_2 = G_1^{C_2}$, . . . , $G_s = G_{s-1}^{C_s}$, где C_i — правильный цикл в G_{i-1}. Обозначим r_i образ корня r при отображении $\varphi^{C_i} \circ \varphi^{C_{i-1}} \circ \ldots \circ \varphi^{C_1}$, а P_i — образ P при том же отображении. Очевидно, r_i — свободная вершина в G_i. В графе G_s имеется чередующееся дерево T с корнем r_s. На очередной итерации производится работа с деревом T, которая заключается в выполнении одной из следующих процедур:

П1. Нахождение ув. чередующейся цепи проводится, если существует ребро (x, v) такое, что $x \in E(T)$, v — свободная вершина ($\neq r_s$). В этом случае $L_{v,r_s} = v, L(x)$ — ув. чередующаяся цепь. По цепи L_{v,r_s} последовательно строятся цепи $L_{v,r_{s-1}} \subset G_{s-1}$, $L_{v,r_{s-2}} \subset G_{s-2}$, . . ., $L_{v,r} \subset \overline{G}$ (для каждого i $L_{v,r_i} = \overline{\varphi}^{C_i}(L_{v,r_{i-1}})$)[3], т. е. в результате в графе G оказывается найденной ув. чередующаяся цепь $L_{v,r}$, и после проведения перекраски вдоль $L_{v,r}$ этап заканчивается. Покажем, как по цепи $L_{v,r_i} \subset G_i$ построить цепь $L_{v,r_{i-1}} \subset G_{i-1}$. Если вершина C_i не принадлежит L_{v,r_i}, то $L_{v,r_i} = L_{v,r_{i-1}}$. Пусть $C_i \in V(L_{v,r_i})$ и (C_i, x) — выделенное, а (y, C_i) — невыделенное ребра в ней. Найдем в G_{i-1} ребро (y, z), где $z \in V(C_i)$ (такое ребро существует в силу существования $(y, C_i) \in U(G_i)$). Цепь $L_{v,r_{i-1}}$ получается из цепи L_{v,r_i} заменой участка y, C_i, x цепью $L_{y,x} = y, L_{z,r(C_i)}^{C_i}, x$ (такую замену будем называть *углублением* в вершине C_i).

П2. Наращивание чередующегося дерева производится, если существует ребро (x, y) такое, что $x \in E(T)$, $y \notin V(T)$ и y — несвободная. В силу свойства в) $p(y) \notin V(T)$. Добавляем к T вершины y и $z = p(y)$ и ребра (x, y) и (y, z). Очевидно, новое дерево — чередующееся и z — внешняя вершина в нем.

П3. Образование нечетного цикла и редукция графа производятся, если найдется ребро (x, y) такое, что $x \in E(T)$, $y \in E(T)$. Пусть z — первая общая вершина в возвратных цепях $L(x)$ и $L(y)$. Объединение участков цепей $L(x)$ от x до z и $L(y)$ — от y до z и ребра (x, y) является правильным циклом (обозначим его C_{s+1}). Производим редукцию графа G_s, т. е. построение графа $G_{s+1} = G_s^{C_{s+1}}$. Деревом в нем будет образ старого дерева при отображении $\overline{\varphi}^{C_{s+1}}$.

Пусть, наконец, получен граф G_s и дерево T, к которым не могут быть применены процедуры П1 — П3 (дерево T в этом случае называется венгерским). Из теории Эдмондса следует:

Теорема 3. В рассматриваемом случае $E(G_s) = E(T)$,

[3] Вершина v, как будет следовать из описания алгоритма, уже является вершиной графа $\overline{G}$.

$I(G_s) = I(T)$ (здесь $E(G_s)$ ($I(G_s)$) — множество внешних (внутренних) относительно корня r_s вершин). ■

Из теорем 2 и 3 следует, что множество $E(I)$ совпадает с прообразом множества $E(T)$ (соответственно $I(T)$) при отображении $\varphi^{C_s} \circ \varphi^{C_{s-1}} \circ \ldots \circ \varphi^{C_1}$. Таким образом, задача $\mathbf{A}_r$ имеет отрицательное решение. Производится процедура

П4. Выделение подграфа $G_{\{V(\bar{G}) \setminus (E \cup I)\}}$. Этап окончен. В дальнейшем в соответствии с теоремой 1 работа алгоритма переносится на подграф $G_{\{V(\bar{G}) \setminus (E \cup I)\}}$.

4. Реализация алгоритма решения задачи $\mathbf{A}_r$. Будем говорить, что множество (быть может, пустое) $\{m_1, m_2, \ldots, m_l\}$ задано списком, если фиксировано некоторое упорядочение $m_{i_1}, m_{i_2}, \ldots, m_{i_l}$, которое определяется указанием адресов начального и конечного элементов и для каждого элемента — адресов предыдущего и последующего (для начального — только последующего, конечного — только предыдущего).

Информация о каждом графе в алгоритме задается списком его вершин и для каждой вершины — списком вершин, смежных с ней (этот список можно считать списком ребер).

В процессе алгоритма вместо последовательности графов $\bar{G}, G_1, G_2, \ldots, G_s$ помнятся только первый — $\bar{G}$ и последний — $\tilde{G} = G_s$; кроме того, имеется набор графов $F_1, F_2, \ldots, F_s$, которые назовем *фрагментами*. Фрагмент F_i представляет собой часть графа G_{i-1}; его вершинами являются вершины цикла C_i и смежные с ними (назовем последние *периферийными*); ребрами являются ребра цикла C_i и ребра, соединяющие вершины из C_i с периферийными вершинами, причем из всех ребер, инцидентных данной периферийной вершине, в F_i присутствует ровно одно. В F_i выделен цикл C_i и его корень $r(C_i)$.

Информация о текущем дереве T задается указанием для каждой некорневой вершины следующей вершины в возвратной цепи.

Если в $\tilde{G}$ есть ребра, соединяющие какую-либо свободную некорневую вершину с вершиной из $E(T)$, то одно из таких ребер указано в ячейке $M_{\text{св}}$. В процессе алгоритма поддерживаются два рабочих списка ребер, концы которых могут либо принадлежать, либо не принадлежать $\tilde{G}$, причем если оба конца принадлежат $\tilde{G}$, то и само ребро принадлежит $\tilde{G}$: а) $M_{\text{нар}}$ — список, в котором в случае $M_{\text{св}} = \{\phi\}$ присутствуют все ребра $(x, y) \in U(\tilde{G})$, где $x \in E(T)$, $y \notin V(T)$; б) $M_{\text{ред}}$ — список, в котором в том же случае присутствуют все ребра $(x, y) \in U(G)$, где $x \in E(T)$, $y \in E(T)$.

В начале этапа в качестве $\tilde{G}$ берется дубликат графа $\bar{G}$. Для того чтобы сделать итерации стереотипными, пополним граф $\tilde{G}$ «фиктивными» вершинами $r_{\text{ф}}$, $r'_{\text{ф}}$ и ребрами $(r_{\text{ф}}, r'_{\text{ф}})$ и $(r'_{\text{ф}}, r)$, второе из которых будем считать выделенным. Вместо задачи $\mathbf{A}_r$ будем решать задачу $\mathbf{A}_{r_{\text{ф}}}$ (заметим, что каждой ув. чередующейся цепи $L_{v,r}$ в $\bar{G}$ взаимно-однозначно соответствует ув. чередующаяся цепь $L_{v,r_{\text{ф}}} = L_{v,r}, r'_{\text{ф}}, r_{\text{ф}}$ в $\tilde{G}$). Вначале положим $M_{\text{св}} = \{\phi\}$, $M_{\text{нар}} = \{(r_{\text{ф}}, r'_{\text{ф}})\}$, $M_{\text{ред}} = \{\phi\}$.

Опишем реализацию алгоритма. На очередной итерации просматривается ячейка $M_{\text{св}}$, и если она непуста, то производится процедура П1. Если $M_{\text{св}} = \{\phi\}$, то идем по списку $M_{\text{нар}}$ с его начала до тех пор, пока не встретим ребро (x, y), принадлежащее графу $\tilde{G}$, причем $x \in E(T)$, $y \notin V(T)$, и тогда осуществляем процедуру П2. Если в $M_{\text{нар}}$ ребра из $\tilde{G}$ не содержатся (т. е. наращивание дерева T невозможно), то просматриваем список $M_{\text{ред}}$ (с его начала). Как только найдено ребро (x, y), принадлежащее графу $\tilde{G}$, осуществляется процедура П3. Если такого ребра в $M_{\text{ред}}$ нет, то производится процедура П4. Все просмотренные элементы из списков $M_{\text{нар}}$ и $M_{\text{ред}}$ исключаем.

Реализация процедуры П1. Пусть (x, v) — ребро, содержащееся в $M_{\text{св}}$, $x \in E(T)$, v — свободная вершина. Выделяем ув. чередующуюся цепь $L_{v,r_{\text{ф}}} = v, L(x)$, где $L(x)$ — возвратная цепь в T. Пусть несколько углублений уже проведено и найдена ув. чередующаяся цепь $\bar{L}_{v,\, r_\Phi}$. Покажем, как произвести следующее углубление. В цепи $\bar{L}_{v,\, r_\Phi}$ отыскиваем вершину C_i с наибольшим номером i (если в $\bar{L}_{v,\, r_\Phi}$ редуцированных вершин нет, то $\bar{L}_{v,\, r_\Phi}$ уже искомая ув. чередующаяся цепь). Пусть x и y — смежные с C_i вершины в цепи $\bar{L}_{v,\, r_\Phi}$ и пусть для определенности ребро $(C_i,\ y)$ — невыделенное. Очевидно, x и y — периферийные вершины в F_i. Находим в F_i цепь $L_{y,\, r(C_i)} = y,\ L^{C_i}_{z,\, r(C_i)}$, где z — смежная с y вершина цикла C_i, и $L^{C_i}_{z,\, r(C_i)}$ — чередующаяся цепь четной длины, лежащая в C_i. Заменяем цепью $L_{y,\, r(C_i)}$, x участок $y,\ C_i,\ x$ в цепи $\bar{L}_{v,\, r_\Phi}$.

Оценим число действий реализации. На построение начальной цепи L_{v, r_Φ} в $\bar{G}$ достаточно $O\ (|\ V\ (L_{v, r_\Phi})\ |)$ действий. Углубление в вершине C_i осуществимо за $O\ (|\ V(L_{y, r(C_i)})\ |)$ действий, и, следовательно, число действий на все углубления, вместе взятые, «пропорционально» длине найденной ув. чередующейся цепи и может быть оценено как $O\ (n)$. Поиск очередной вершины, в которой надо произвести углубление, очевидно, осуществим за $O\ (n)$ действий, и, следовательно, число таких действий за всю процедуру оценивается как $O\ (n^2)$ [4]. Таким образом, оценка числа действий процедуры — $O\ (n^2)$ (с учетом замечания в сноске процедура может быть осуществлена с оценкой числа действий $O\ (n \cdot \log_2 n)$).

Реализация процедуры П2. Пусть $(x,\ y)$ — ребро графа $\bar{G}$, найденное при работе со списком $M_{\text{нар}}$, $x \in E\ (T)$, $y \notin V\ (T)$. Реализация состоит из следующих частей:

а) Нахождение вершины $z = p\ (y)$ и присоединение к T вершин y и z и ребер $(x,\ y)$ и $(y,\ z)$. Оценка числа действий — $O\ (1)$.

[4] В статье «Справочная для выборки максимального элемента и ее приложения» в настоящем сборнике описывается метод, применяя который к поиску вершины C_i с максимальным номером i можно получить суммарную оценку числа действий $O\ (n \cdot \log_2 n)$.

б) Пополнение списков $M_{\text{нар}}$ и $M_{\text{ред}}$. Последовательно просматриваем ребра в $\bar{G}$, инцидентные z. Пусть $(z,\ t)$ — очередное ребро. Если t — свободная вершина, то заносим $(z,\ t)$ в ячейку $M_{\text{св}}$ и итерацию оканчиваем. Если вершина t — несвободная и $t \notin V\ (T)$, то заносим $(z,\ t)$ в список $M_{\text{нар}}$ (из эвристических соображений целесообразнее заносить $(z,\ t)$ в начало списка $M_{\text{нар}}$). Если $t \in E\ (T)$, то помещаем $(z,\ t)$ в список $M_{\text{ред}}$ (также в его начало). Оценка числа действий — $O\ (|\ U\ (z)\ |)$, где $U\ (z)$ — множество ребер, инцидентных z.

Реализация процедуры П3. Пусть $(x,\ y)$ — ребро графа $\bar{G}$, найденное при работе со списком $M_{\text{ред}}$, $x \in E\ (T)$, $y \in E\ (T)$. Реализация состоит из следующих частей:

а) Выделение цикла C_{s+1} (s — номер последней редукции). Поочередно идем по возвратным цепям $L\ (x)$ и $L\ (y)$ и помечаем пройденные вершины. Останавливаемся, как только появилась вершина, помеченная дважды (она, очевидно, и будет корнем $r\ (C_{s+1})$). Оценка числа действий— $O\ (|\ V\ (C_{s+1})\ |)$.

б) Построение F_{s+1}. Включаем в F_{s+1} цикл C_{s+1}. Последовательно просматриваем вершины цикла. Просматривая вершину $x \in V\ (C_{s+1})$, перебираем вершины, с ней смежные. Если вершина y, смежная вершине x, не принадлежит C_{s+1} и еще не включена в F_{s+1}, то вводим ее в F_{s+1} вместе с ребром $(x,\ y)$. Число действий на построение F_{s+1} «пропорционально» числу ребер, инцидентных вершинам из C_{s+1}, и может быть оценено как $O\ (|\ V\ (C_{s+1})\ | \cdot n)$.

в) Редукция графа $\bar{G}$. «Новый» граф $\bar{G}$ получается из «старого» исключением вершин из C_{s+1} вместе с инцидентными им ребрами и добавлением вершины C_{s+1} и производных ребер. Эта процедура может быть проделана параллельно с построением F_{s+1}. Число действий оценивается как $O\ (|\ V\ (C_{s+1})\ | \cdot n)$.

г) Пополнение списков $M_{\text{нар}}$ и $M_{\text{ред}}$. Процедура осуществляется просмотром дуг нового графа $\bar{G}$, инцидентных вершине C_{s+1}, как и для вершины z в реализации процедуры П2 (часть б).

Реализация процедуры П4. Вершины, подлежащие удалению,— это вершины из $I(T)$, вершины из $E(T)$, принадлежащие $\overline{G}$, а также вершины фрагментов, принадлежащие $\overline{G}$. Число действий на исключение этих вершин, а также инцидентных им дуг может быть оценено как $O(n^2)$.

Легко доказать по индукции, что соответствующие требования, накладываемые на ячейку $M_{\text{св}}$ и списки $M_{\text{нар}}$ и $M_{\text{ред}}$, всегда выполняются.

Оценка числа действий и рабочей памяти. Как отмечалось в п. 3, число этапов не превосходит n. Оценим число действий реализации на этапе.

Лемма 1. $\sum_{i=1}^{k} |V(C_i)| < \frac{3}{2}n$ (здесь k — число редукций на этапе).

Доказательство. При редукции с циклом C_i число вершин в графе $\tilde{G}$ уменьшается на $|V(C_i)| - 1$, и, поскольку $|V(C_i)| \geqslant 3$, то $k < n/2$. Обозначим через $\bar{n}$ число нефиктивных вершин графа $\tilde{G}$ в конце этапа. Имеем

$$\bar{n} \leqslant n - \sum_{i=1}^{k} (|V(C_i)| - 1),$$

откуда $\sum_{i=1}^{k} |V(C_i)| \leqslant n - \bar{n} + k < \frac{3}{2}n$. ■

Используя лемму 1 и уже полученные оценки реализаций отдельных процедур, можно получить оценку числа действий на этапе $O(n^2)$ (заметим, что на поиск нужных элементов в списках $M_{\text{нар}}$ и $M_{\text{ред}}$ тратится число действий, «пропорциональное» общему числу элементов, побывавших в этих списках, т. е. оценивается через число действий на пополнение этих списков в реализациях процедур П2 и П3).

Рабочая память расходуется на хранение графов $\overline{G}$, $\tilde{G}$, фрагментов и списков $M_{\text{нар}}$ и $M_{\text{ред}}$, а также структур внутри графов. Число всех вершин в фрагментах на основании леммы 1 не более $^3/_2 n$, следовательно, число дуг в них $O(n^2)$. Число элементов в списках $M_{\text{нар}}$, $M_{\text{ред}}$ также оценивается как $O(n^2)$. Следовательно, оценка памяти — $O(n^2)$.

5. Задача Б. Следуя [2], поставим в соответствие каждому ребру $u \in U(G)$ действительную переменную α_u.

Рассмотрим следующую задачу L линейного программирования:

$$\alpha_u \geqslant 0, \qquad \forall u \in V(G), \tag{1}$$

$$\sum_{u \in U(v)} \alpha_u \leqslant 1, \qquad \forall v \in V(G)^5, \tag{2}$$

$$\sum_{u \in U(G_{\{S\}})} \alpha_u \leqslant k_S, \qquad \forall S \subset V(G) : |S| = 2k_S + 1,$$

$$k_S \text{ — положительное целое,} \tag{3}$$

$$L(\alpha) = (\mathbf{c}, \boldsymbol{\alpha}) = \sum_{u \in U} (c(u), \alpha_u) \to \max.$$

Каждому паросочетанию P соответствует план $\boldsymbol{\alpha}^P$ задачи L: $\alpha_u = 1 \Leftrightarrow u \in P$, $\alpha_u = 0 \Leftrightarrow u \notin P$, который также будем называть паросочетанием.

Отнесем действительную переменную β_v (γ_s) соответствующей строке в условиях (2) и (3). Двойственная к L задача L^* имеет вид:

$$\beta_v \geqslant 0, \qquad \gamma_S \geqslant 0; \tag{4}$$

$$\beta_{v'} + \beta_{v''} + \sum_{S \subset V(G) : (v', v'') \in U(G_{\{S\}})} \gamma_S \geqslant c(v', v''),$$

$$\forall (v', v'') \in U(G); \tag{5}$$

$$L^*(\beta, \gamma) = \sum_{v \in V(G)} \beta_v + \sum_{S \subset V(G) : |S| = 2k_S + 1} k_S \gamma_S \to \min.$$

[5] $U(v)$ обозначает множество ребер, инцидентных v.

Будем называть β_v потенциалом вершины v, γ_S — добавкой множества S.

Если α — план задачи L и β, γ — план задачи L^*, то, как известно, $L(\alpha) \leqslant L^*(\beta, \gamma)$. В результате алгоритма [2] оказываются построенными паросочетание M и план β, γ задачи L^*, для которых

$$L(\alpha^M) = L^*(\beta, \gamma),$$

откуда следует, что M—паросочетание максимального веса.

Таким образом, теоретическим результатом алгоритма будет:

Теорема 4 [2]. Паросочетание максимального веса является оптимальным решением задачи L. Опорными планами задачи L являются паросочетания. ■

Пусть для некоторых паросочетания P и плана β, γ задачи L^* выполняются следующие соотношения:

Если $v \in V(G)$ — свободная вершина, то $y_v = 0$. (6)

Для любого выделенного ребра условие (5) обращается в равенство. (7)

Если $\gamma_S > 0$, то в $G_{\{S\}}$ ровно k_S ребер из P. (8)

Можно убедиться, что соотношения (6) — (8) — условия дополняющей нежесткости (см., например, [6], стр. 176) для планов α^P и β, γ. Из известной теоремы (см. теорему 7.2 там же, стр. 178) следует, что при этом эти планы оптимальны. В алгоритме как раз конструируются паросочетание и план задачи L^*, подчиняющиеся соотношениям (6) — (8).

6. **Алгоритм Эдмондса решения задачи Б.** Мы изложим алгоритм в модифицированном виде.

Примем ряд определений и обозначений. Пусть в $V(G)$ имеется совокупность подмножеств R такая, что: а) в R содержатся все одновершинные подмножества; б) каждое подмножество $x \in R$ содержит нечетное $(2k_x + 1)$ число вершин; в) для любых различных подмножеств x, $y \in R$ либо $x \subset y$, либо $y \subset x$, либо $x \cap y = \phi$. Все подмножества из R будем называть вершинами. Одновершинные подмножества назовем простыми вершинами. Вершина $y \in R$ называется *субвершиной* вершины $x \in R$, если $y \subset x$ и не существует $z \in R$ такого, что $y \subset z \subset x$. Множество субвершин вершины x обозначим $s(x)$. Число субвершин у каждой непростой вершины x, очевидно, нечетное; обозначим его $2\bar{k}_x + 1$. *Субграфом* $G^{s(x)}$ непростой вершины $x \in R$ назовем факторграф графа $G_{\{x\}}$, множеством вершин которого является $s(x)$. Субграфом простой вершины будем по определению считать саму вершину.

Вершина из R называется главной, если она не содержится внутри никакой другой вершины. Множество главных вершин обозначим $R_{гл}$. Зададим естественное отображение $\varphi : V(G) \to R_{гл}$. Факторграф графа G, множеством вершин которого является $R_{гл}$, обозначим G_R.

Пусть P — паросочетание в G и пусть для каждого множества $x \in R$ в графе $G_{\{x\}}$ содержится ровно k_x ребер из P. Вершину, не инцидентную выделенным ребрам, лежащим в $G_{\{x\}}$, обозначим $r(x)$. Обозначим P_R образ выделенных ребер при отображении $G \to G_R$ и $P^{s(x)}$ — образ выделенных ребер при отображении $G_{\{x\}} \to G^{s(x)}$, где $x \in R$. Справедлива

Лемма 2. Множества P_R и $P^{s(x)}$ являются паросочетаниями в соответствующих графах. $|P^{s(x)}| = \bar{k}_x$. ■

Совокупность R при паросочетании P назовем *системой*, если для каждого $x \in R$ в $G^{s(x)}$ имеется чередующийся цикл $C^{s(x)}$ (относительно паросочетания $P^{s(x)}$), содержащий все его $2\bar{k}_x + 1$ вершин.

Для каждой непростой вершины $x \in R$ обозначим $r(C^{s(x)})$ корень цикла $C^{s(x)}$.

Пусть β, γ — текущий план задачи L^*. Ребро $u \in V(G)$ назовем допустимым, если для него (5) обращается в равенство. Обозначим $G^д$ — текущий суграф графа G [6], содержащий допустимые ребра, и только их.

[6] То есть $V(G^д) = V(G)$, $U(G^д) \subseteq U(G)$.

Описание алгоритма. Вначале полагаются: а) $\gamma_S = 0$, $\forall\, S \subset V$; б) достаточно большие потенциалы вершин, с тем чтобы выполнялись (4) и (5); в) $P = \phi$.

В алгоритме производятся многократные изменения текущих паросочетания, потенциалов и добавок. Соотношения (7) и (8) будут выполняться всегда, соотношение (6) может нарушаться. Число свободных вершин, для которых (6) нарушается, будет монотонно уменьшаться до нуля; следовательно, в результате будет построено паросочетание максимального веса. Работу алгоритма между двумя уменьшениями назовем этапом.

Пусть уже проведено несколько этапов и в текущем суграфе $G^{д}$ найдено паросочетание P, а также в $G^{д}$ имеется система R. Пусть добавки отличны от нуля только, быть может, для подмножеств, являющихся непростыми вершинами в R. Считаем, что для каждого множества $x \in R$ вычислена величина $\beta_x = \min\limits_{y \in V(G),\, y \in x} \beta_y$, которую назовем потенциалом вершины x.

На очередном этапе в графе G выбирается произвольная свободная вершина r, для которой $\beta_r > 0$ (если таких вершин нет, то P — паросочетание максимального веса). Пусть $\tilde{r}$ — главная вершина, которой принадлежит r. Строим в рабочем графе $G_R^{д}$ чередующееся дерево T с корнем $\tilde{r}$.

П'1. Нахождение ув. чередующейся цепи. По найденной ув. чередующейся цепи $L_{v,\tilde{r}}$ в $G_R^{д}$ путем ряда углублений находится ув. чередующаяся цепь $L_{r(v),\,r}$ в $G^{д}$ $(L_{v,\tilde{r}} = \varphi\,(L_{r(v),\,r}))$.

При перекраске вдоль $L_{r(v),\,r}$ число свободных вершин в G с потенциалом больше нуля уменьшается на одну или две. Этап считается законченным. Заметим, что для паросочетания, полученного в результате перекраски, продолжают выполняться соотношения (7) и (8), а также сохраняется цикл $C^{s(x)}$, $\forall\, x \in R$ (быть может, меняется его корень), т. е. R продолжает оставаться системой.

П'2. Наращивание чередующегося дерева. Процедура проводится аналогично П2.

П'2'. Нахождение чередующейся цепи, ведущей во внешнюю вершину с нулевым потенциалом. При очередном наращивании (процедура П'2) в дереве T может появиться вершина $\tilde{v}$ с нулевым потенциалом. Тогда существует вершина $v \in V\,(G)$ такая, что $\tilde{v} = \varphi\,(v)$ и $\beta_v = 0$. Подобно процедуре П'1 по возвратной цепи $L\,(\tilde{v})$ в T строится чередующаяся цепь $L_{v,\,r}$ в $G^{д}$ $(L\,(\tilde{v}) = \overline{\varphi}\,(L_{v,\,r}))$. Производится перекраска вдоль $L_{v,\,r}$, при этом вершина v становится свободной, а r — несвободной. Таким образом число свободных вершин с ненулевым потенциалом уменьшается на единицу. Этап считается оконченным.

П'3. Образование цикла и редукция графа. Процедура аналогична П3. В систему R включается новое множество $x = \varphi^{-1}\,(V\,(C))$, для которого полагается $\gamma_x = 0$ (здесь C — найденный цикл). Очевидно, условие (8) для x выполняется.

Пусть, наконец, получено дерево T, к которому неприменимы процедуры П'1, П'2, П'2', П'3 (венгерское дерево). Будем «непрерывно уменьшать» потенциалы во внешних вершинах T и увеличивать во внутренних на одно и то же число. Уменьшая (увеличивая) на ε потенциал вершины $x \in V\,(T)$, будем уменьшать (увеличивать) на ε потенциал каждой вершины $y \in R$, такой, что $y \in x$. Кроме того, если указанная вершина x непростая, будем увеличивать (уменьшать) на 2ε добавку γ_x. Очевидно, при этом допустимость ребер графа G, оба конца которых принадлежат одной и той же главной вершине либо один конец принадлежит внешней, а другой — внутренней в T главной вершине, сохраняется. Заметим, что, если $\varepsilon > 0$ ребро $(x,\ y) \in U\,(G^{д})$ такое, что $\varphi\,(x) \in I\,(T)$, $\varphi\,(y) \in I\,(T)$ (такое ребро назовем внутренним) перестает быть допустимым, хотя для него продолжает выполняться (5). Тем самым внутренние ребра следует удалить из $G^{д}$, а их производные — из $G_R^{д}$.

Перестаем «наращивать» ε, как только имеет место один из следующих случаев:

С1. Для некоторой вершины $v \in E(T)$ стало $\beta_v = 0$. Производим процедуру П′2′ и этап считаем законченным.

С2. Для некоторого ребра $(x, y) \in U(G) : \varphi(x) \in E(T)$, $\varphi(y) \in E(T)$ (такое ребро назовем внешним) (5) обратилось в равенство (т. е. ребро (x, y) стало допустимым). Пополняем граф $G^{д}$ новыми допустимыми ребрами, а граф $G_R^{д}$ — их производными. Дерево T при этом перестает быть венгерским и можно производить редукцию графа (процедура П′3).

С3. Некоторое ребро $(x, y) \in U(G)$: $\varphi(x) \in E(T)$, $\varphi(y) \notin V(T)$ (такое ребро назовем полувнешним) стало допустимым. Как и в случае С2, пополняем граф $G^{д}$ такими ребрами, а граф $G_R^{д}$ — их производными. К новому графу $G_R^{д}$ применима процедура П′1, П′2 или П′2′.

С4. Для некоторой непростой вершины $x \in I(T)$ стало $\gamma_x = 0$. В этом случае множество x перестаем считать элементом системы R и производим соответствующие исправления в графе $G_R^{д}$ и дереве T. Образно говоря, в граф $G_R^{д}$ на место вершины x «вклеивается» суграф $G^{s(x)}$. В T вершине x были инцидентны ровно два ребра (пусть это выделенное ребро (x, y) и невыделенное (x, z)). В новом графе $G_R^{д}$ должны существовать выделенное ребро $(r(C^{s(x)}), y)$ и невыделенное — (x', z), $x' \in s(x)$. Новое дерево T получается добавлением к части старого дерева, расположенной на неизменившейся части графа $G_R^{д}$, чередующейся цепи $L_{y,z} = y$, $L^{C^{s(x)}}_{r(C^{s(x)}),\, x'}$, z.

Сходимость алгоритма. Докажем конечность работы на этапе. Рассмотрим множества $\tilde{E} = \varphi^{-1}(E(T)) \subset V(G)$ и $\tilde{I} = \varphi^{-1}(I(T)) \subset V(G)$. Справедлива

Лемма 3. Каждая вершина может войти в множество $\tilde{E}$ ($\tilde{I}$) не более одного раза.

Можно проверить, что процедуры П′2, П′3, С4, а также С2 в сочетании с П′3 и С3 в сочетании с П′2 расширяют множество $\tilde{E}$, откуда с учетом леммы 3 следует, что общее

число этих процедур не более $2n$. Тем самым конечность этапа, а вместе с ним и алгоритма доказана.

7. Реализация этапа алгоритма. Основными объектами работы будут граф $G_R^{д}$ с деревом T и глобальный граф $\mathfrak{G} = [R, \mathfrak{U}]$. Множество вершин графа $\mathfrak{G}$ изоморфно R. В граф $\mathfrak{G}$ включены все ребра графа G и ребра вида (x, y), $x \in R$, $y \in R$, если существует ребро (x', y'), где $x' \in V(G)$, $y' \in V(G)$, $x' \in x$, $y' \in y$ (ребро (x, y) назовем производным от ребра (x', y'), а также каждое ребро (x, y), где $x, y \in R$ и $y \in s(x)$; такое ребро (x, y) назовем *особым* и зададим на нем направление от x к y. Все ребра, производные от выделенных, также будем считать выделенными. В $\mathfrak{G}$ отмечен чередующийся цикл $C^{s(x)}$ для каждой непростой вершины $x \in R$. Граф $G^{д}$ естественно вложен в граф $\mathfrak{G}$. Дефектом $d(x, y)$ ребра $(x, y) \in U(G)$ при плане β, γ назовем величину $\beta_x + \beta_y + \sum\limits_{S \subset V(G):(x, y) \in U(G_{\{S\}})} \gamma_s - c(x, y)$ (дефект допустимых дуг равен нулю). Будем считать помеченным текущие множества $\tilde{E}$ и $\tilde{I}$, устроенные в виде списков.

В процессе работы поддерживаются текущее множество B внутренних ребер и две неоднородные справочные (описание неоднородной справочной и работы с ней дано в статье «Справочная для выборки максимального элемента и ее приложения» из настоящего сборника): 1) справочная $Q_в$ внешних ребер; 2) справочная $Q_{пв}$ полувнешних ребер. Для справочной $Q_в$ ($Q_{пв}$) указана *добавка* $\Delta_в$ ($\Delta_{пв}$) [7]. Каждому ребру $(x, y) \in Q_в$ отнесено число $a(x, y) = d(x, y) + \Delta_в$ (аналогично в $Q_{пв}$). Элементы в справочных $Q_в$ и $Q_{пв}$ расположены по возрастанию a.

Как и для реализации алгоритма решения задачи **А**, имеются списки $M_{нар}$, $M_{ред}$ и ячейка $M_{св}$.

Опишем реализации процедур алгоритма и одновременно установим для них оценки числа действий.

---

[7] Смысл добавок выяснится в дальнейшем.

1) **Реализация процедуры П′1.** Опишем процедуру углубления на примере первого углубления. В ув. чередующейся цепи $L_{v,\tilde{r}}$ находится произвольная непростая вершина x. Пусть она инцидентна в $L_{v,\tilde{r}}$ выделенному ребру (x, y) и невыделенному (x, z). При помощи особых ребер находим цикл $C^{s(x)}$ и его корень $r\,(C^{s(x)})$. Перебирая в $\mathfrak{G}$ ребра, инцидентные z, найдем ребро (z, x'), где $x' \in s\,(x)$. «Вклеиваем» цепь $L_{z,y} = z,\ L^{C^{s(x)}}_{x',\ r\,(C^{s(x)})},\ y$ в соответствующее место цепи $L_{v,\tilde{r}}$. Оценка числа действий на одно углубление — $O\,(n)$ и, следовательно, на поиск искомой ув. чередующейся цепи — $O\,(n^2)$.

2) **Реализация процедуры П′2** в основной части совпадает с реализацией процедуры П2. Пусть y — новая внутренняя, а z — внешняя вершины дерева T. Используя особые ребра графа $\mathfrak{G}$, находим множества $\varphi^{-1}\,(y)$ и $\varphi^{-1}\,(z)$, которыми пополняем множества $\tilde{T}$ и $\tilde{E}$; соответственно перебирая ребра графа G, инцидентные вершинам множеств $\varphi^{-1}\,(y)$, $\varphi^{-1}\,(z)$, находим те из них, которыми нужно пополнить множество B и справочные $Q_{\text{в}}$ и $Q_{\text{вп}}$. Для каждого ребра u, попадающего в $Q_{\text{в}}$ $(Q_{\text{вп}})$, вычисляем $a\,(u) = d\,(u) + \Delta_{\text{в}}$ $(a\,(u) = d\,(u) + \Delta_{\text{пв}})$. Оценка числа действий рассматриваемой реализации без учета внесения найденных ребер в справочные $Q_{\text{в}}$ и $Q_{\text{пв}}$[8] — $O\,(n \cdot |\,\varphi^{-1}\,(y) + \varphi^{-1}\,(z)\,|)$ и в силу леммы 3 оценка за этап — $O\,(n^2)$.

3) **Реализация процедуры П′2′** схожа с реализацией процедуры П′1. Отличие состоит в том, что при углублении в начальной вершине x чередующейся цепи требуется найти новое начало — вершину $x' \in s\,(x)$, для которой $\beta_{x'} = 0$.

4) **Реализация процедуры П′3.** Нахождение чередующегося цикла C и редукция по нему графа $G^{\text{д}}_{\text{R}}$ проводятся, как и в реализации П3. Вместо построения фрагмента производится достройка графа $\mathfrak{G}$: к нему присоединяется новая вершина x и добавляются: а) ребра, соединяющие x со всеми вершинами в C (особые ребра); б) ребра вида (x, y), если существует ребро (x', y), где $x' \in V\,(C)$. Как и при реализации процедуры П′2, надо позаботиться о пополнении множества $\tilde{E}$, а также об удалении ряда вершин из $\tilde{T}$. Наряду с этим надо пополнить соответствующими элементами справочные $Q_{\text{в}}$ и $Q_{\text{пв}}$ и уменьшить список B. Для этого находим множество $I\,(C)$ вершин цикла C, бывших внутренними для прежнего дерева, и определяем множество $\varphi^{-1}\,(I\,(C))$. Полагаем $\tilde{E} := \tilde{E} \bigcup \varphi^{-1}\,(I\,(C))$, $\tilde{T} := \tilde{T} \setminus \varphi^{-1}\,(I\,(C))$. Оценка числа действий процедуры без учета внесения найденных дуг в справочные $Q_{\text{в}}$ и $Q_{\text{пв}}$ — $O\,(n \cdot |\,\varphi^{-1}\,(I\,(C))\,|)$ и на основании леммы 3 за этап — $O\,(n^2)$.

8 Эти действия будут учтены особо.

5) **Работа с венгерским деревом.** Находятся следующие элементы: а) $\tilde{x} \in I\,(T)$, для которого $\gamma_{\tilde{x}}$ минимально; б) $\tilde{\tilde{x}} \in E\,(T)$, для которого $\beta_{\tilde{\tilde{x}}}$ минимально; в) $Q \subset Q_{\text{в}}$ (соответственно $\tilde{\tilde{Q}} \subset Q_{\text{пв}}$) — подмножество ребер с минимальным дефектом. Множества $\tilde{Q}$ и $\tilde{\tilde{Q}}$ отыскиваются как старшие элементы в справочных (с одинаковым значением α), $\tilde{x}$ и $\tilde{\tilde{x}}$ можно найти перебором вершин в T.

Определение, какую из процедур С1 — С4 нужно производить, зависит от того, какая из величин $\frac{\gamma_{\tilde{x}}}{2}$, $\beta_{\tilde{\tilde{x}}}$, $d\,(\tilde{u})/2$, $d\,(\tilde{\tilde{u}})$ (где $\tilde{u} \in \tilde{Q}$, $\tilde{\tilde{u}} \in \tilde{\tilde{Q}}$) минимальна. Минимум этих величин обозначим ε. Увеличим на 2ε и ε добавки $\Delta_{\text{в}}$ и $\Delta_{\text{пв}}$ соответственно (с тем, чтобы новый дефект ребра u по-прежнему равнялся $a\,(u) - \Delta\,(u)$).

Если $\varepsilon > 0$, то все внутренние ребра перестают быть допустимыми. Выкидываем ребра B из графа $G^{\text{д}}$, а также производные от них ребра в $\mathfrak{G}$. Производим выкидывание таких ребер и из $G^{\text{д}}_{\text{R}}$. Список B обнуляем. В силу леммы 3 число таких действий за этап — $O\,(n^2)$.

Пересчитываем потенциалы вершин графа $G^{\text{д}}$, после чего, используя особые ребра, пересчитываем потенциалы вершин графа $\mathfrak{G}$. Определяем новые значения добавок в главных непростых вершинах. Оценка числа действий этих операций — $O(n)$, а за весь этап — $O(n^2)$.

Реализация процедуры С1 проводится так же, как и процедуры П′2′.

Реализация процедуры С2(С3). Добавляем в граф $\mathfrak{G}$ все ребра из $\widetilde{Q}$ $(\widetilde{\widetilde{Q}})$ и, используя особые ребра, добавляем все производные от них. Вводим соответствующие ребра в граф $G_R^{д}$. Поскольку каждое ребро при этих процедурах может добавиться в граф $\mathfrak{G}$ не более одного раза за этап, оценка числа действий в рассматриваемой реализации за этап — $O(n^2)$.

Реализация процедуры С4. Для непростой вершины $\tilde{x} \in I(T)$ стало $\gamma_{\tilde{x}} = 0$. Удаляем вершину $\tilde{x}$ и инцидентные ей ребра из графов $\mathfrak{G}$ и $G_R^{д}$. Предварительно, пользуясь особыми ребрами, находим множество $s(\tilde{x})$ и добавляем его к $V(G_R^{д})$. Перебирая в графе $\mathfrak{G}$ неособые ребра, инцидентные вершинам из $s(\tilde{x})$, найдем те ребра, оба конца которых принадлежат $G_R^{д}$, и пополним $G_R^{д}$ этими ребрами. Пользуясь отмеченным циклом $C^{s(x)}$, достроим дерево T. Оценка числа действий реализации — $O(n \cdot |s(\tilde{x})|)$, а за весь этап — $O(n^2)$.

Осталось рассмотреть работу со справочными $Q_в$ и $Q_{пв}$. Заметим, что каждое ребро графа G становится внешним (полувнешним) не более одного раза, следовательно, суммарное число элементов, побывавших в $Q_в$ $(Q_{пв})$ за этап, не более p. Тогда, как показано в статье «Справочная для выборки максимального элемента и ее приложения» из настоящего сборника, число действий на поддержание справочной за всю работу оценивается как $O(p \cdot \log_2 p)$, или $O(p \cdot \log_2 n)$.

Итак, оценка реализации алгоритма решения задачи Б за этап — $O(n^2 + p \cdot \log_2 n)$ действий, а за все этапы — $O(n^3 + np \cdot \log_2 n)$. Требуемая рабочая память — $O(n^2)$, поскольку число вершин графа $\mathfrak{G}$ равно $n + k$ (k — число непростых вершин) и $k < n/2$ (см. лемму 1).

ЛИТЕРАТУРА

1. *J. Edmonds.* Paths, trees and flowers.— «Can. J.», Math., 1965, v. 17, N 3.
2. *J. Edmonds.* Maximum matching and polyhedron with 0,1-vertices.— «J. of Research», 1965, v. 69B, No 1, 2.
3. *C. Witzgall, C. Zahn.* Modification of Edmonds maximum matching algorithm.— «J. of Research», 1965, v. 69B, No 1, 2.
4. *V. Balinsky.* Labelling to obtain a maximum matching. Combinatorial Mathematics and its Applications.— «Proc. Conf., Univ. North Carolina, Chapel Hill, N. C., 1967». Univ. North Carolina Press, Chapel Hill, N. C., 1969.
5. *К. Берж.* Теория графов и ее применения. М., ИЛ, 1962.
6. *Д. Б. Юдин, Е. Е. Гольштейн.* Линейное программирование.— «Теория, методы и приложения». М., «Наука», 1969.

А. Я. Гордон

ОДИН АЛГОРИТМ РЕШЕНИЯ МИНИМАКСНОЙ ЗАДАЧИ О НАЗНАЧЕНИИ

1. Известна следующая задача (см., например, [1], стр. 89). В $(n \times n)$-матрице (h_{ij}) требуется найти набор из n независимых клеток (т. е. клеток, стоящих в попарно различных строках и столбцах), для которого максимум выбранных элементов h_{ij} принял бы наименьшее значение (задача I) [1].

К этой задаче сводится, например, следующая «задача о конвейере». Имеется конвейер с n рабочими местами и n рабочими. i-й рабочий может работать на j-м месте при скорости конвейера $\leqslant v_{ij}$. Каким образом следует расставить рабочих по местам, чтобы конвейер можно было пустить с максимальной скоростью? Мы окажемся в ус-

[1] Решение этой и всех далее формулируемых задач, вообще говоря, не единственно. Требуется найти какое-нибудь решение.