

$\in \mathcal{P}_i(x^i)$  как только  $u \in \mathbf{R}_+^l$  и  $0 < \|u\| < \delta(x^i)$ . Учитывая, что в условиях локальной монотонности равновесные цены строго положительны, на основании установленных результатов получаем следующую характеристику вполне договорных состояний.

**Следствие 3.3.** *Пусть модель  $\mathcal{E}$  удовлетворяет условиям теоремы 3.3 или следствия 3.2. Если отношение предпочтения кого-либо из участников  $\mathcal{E}$  локально-монотонно, то  $D_0(\mathcal{E}) = W(\mathcal{E})$ .*

#### ЛИТЕРАТУРА

1. Нейман Дж. фон, Моргенштерн О. Теория игр и экономическое поведение.— М.: Наука, 1970.— 707 с.
2. Ауман Р., Шепли Л. Значения для неатомических игр.— М.: Мир, 1977.— 357 с.
3. Розенмюллер И. Кооперативные игры и рынки.— М.: Мир, 1974.— 167 с.
4. Васильев В. А. Об одном классе дележей в кооперативных играх // Докл. АН СССР.— 1981.— Т. 256, № 2.— С. 265—268.
5. Макаров В. Л. Экономическое равновесие: существование и экстремальные свойства // Современные проблемы математики.— 1982.— Т. 19.— С. 23—59.
6. Васильев В. А. Модели экономического обмена и кооперативные игры.— Новосибирск: Новосибирский ун-т, 1984.— 96 с.
7. Harsanyi J. A. A bargaining model for the cooperative  $n$ -person game // Ann. math. studies.— 1959.— N 40.— P. 325—355.
8. Васильев В. А. Об одном классе операторов в пространстве регулярных функций множества // Оптимизация.— 1982.— Вып. 28.— С. 102—111.
9. Shapley L. S. Cores of convex games // Int. J. game theory.— 1971.— V. 1.— P. 12—26.
10. Dubey P., Neyman A., Weber R. J. Value theory without efficiency // Math. oper. res.— 1981.— V. 6.— P. 122—128.
11. Макаров В. Л., Васильев В. А., Козырев А. Н., Маракулин В. М. О некоторых проблемах и результатах современной математической экономики // Оптимизация.— 1982.— Вып. 30.— С. 5—86.
12. Green J. R. Stability of Edgeworth's recontracting process // Econometrica.— 1974.— V. 42.— P. 21—34.
13. Mashler M., Peleg B. Stable sets and stable points of set-valued dynamics systems with applications to game theory // SIAM J. contr. and optim.— 1976.— V. 14, N 6.— P. 985—995.
14. Васильев В. А. Игра Лукаса не имеет НМ-решений в  $H$ -дележах // Оптимизация.— 1981.— Вып. 27.— С. 5—20.
15. Васильев В. А., Жоробков Р. О. Достижимость ядер классических кооперативных игр // Оптимизация.— 1985.— Вып. 35.— С. 121—133.
16. Hildenbrand W. Core and equilibria in a large economy.— Princeton: Princeton Univ. Press, 1974.— 252 p.

## О НЕКОТОРЫХ МАТЕМАТИЧЕСКИХ МОДЕЛЯХ И МЕТОДАХ ПЛАНИРОВАНИЯ КРУПНОМАСШТАБНЫХ ПРОЕКТОВ

Э. Х. ГИМАДИ

Разработка и исследование ряда крупномасштабных проектов, реализуемых в народном хозяйстве, потребовали изучения таких математических моделей, как задача календарного планирования в условиях ограниченных ресурсов и заданных директивных сроков, задача размещения предприятий на сетях транспортного типа, задача выбора типа оборудования и т. п. Особую актуальность эти вопросы приобретают при реализации крупномасштабных программ в районах нового освоения, в задачах хозяйственного освоения зоны БАМ, при планировании Западно-Сибирского нефтегазового комплекса и др. [1—4].

Указанные математические модели приводят к необходимости рассмотрения так называемых труднорешаемых задач комбинаторной оптимизации [5]. Для этих задач оказывается проблематичным построение эффективных точных алгоритмов решения, трудоемкость которых ограничена полиномом от длины записи входных данных. Попытки построе-

ния малотрудоёмких алгоритмов связаны, как правило, с рассмотрением более узких классов задач, с учетом специфики задач, а также с отказом от требования точности алгоритма. В последнем случае речь идет о построении приближенных алгоритмов с оценками [6—11].

В настоящей статье приводится определение алгоритмов с оценками и асимптотически точных алгоритмов с примерами некоторых результатов для известной задачи коммивояжера (ЗК). Большое внимание уделено одной из самых общих задач теории расписаний: задаче календарного планирования в условиях ограниченных ресурсов и заданных директивных сроков, а также связанным с нею задачам упаковки предметов в контейнеры и в полосу. Кроме того, построены алгоритмы с оценками для решения задач, сходных по своей математической постановке: задачи размещения предприятий на сетях транспортного вида и задачи стандартизации (выбора типажа и состава оборудования — машин, механизмов и т. п.).

### § 1. АЛГОРИТМЫ С ОЦЕНКАМИ И НЕКОТОРЫЕ РЕЗУЛЬТАТЫ ДЛЯ ЗАДАЧИ КОММИВОЯЖЕРА

Наряду с такими важнейшими показателями качества точных алгоритмов, как трудоёмкость (количество элементарных операций)  $T_{\mathcal{A}}$  и память  $\Pi_{\mathcal{A}}$ , необходимыми для реализации алгоритма  $\mathcal{A}$ , для сравнительной оценки приближенных алгоритмов приходится привлекать дополнительные характеристики. Будем говорить, что алгоритм  $\mathcal{A}$  имеет оценки  $\varepsilon_{\mathcal{A}}$ ,  $\delta_{\mathcal{A}}$ , если  $P\{L_{\mathcal{A}} \geq (1 + \varepsilon_{\mathcal{A}}) L^*\} \leq \delta_{\mathcal{A}}$ , где  $P\{S\}$  — вероятность события  $S$ ,  $L_{\mathcal{A}}^*$  — значение целевой функции, даваемое алгоритмом  $\mathcal{A}$ ;  $L^*$  — точное значение целевой функции (в задаче на минимум);  $\varepsilon_{\mathcal{A}}$  — относительная погрешность вычисления целевой функции алгоритмом  $\mathcal{A}$ ;  $\delta_{\mathcal{A}}$  — вероятность несрабатывания алгоритма (доля случаев, когда алгоритм  $\mathcal{A}$  на рассматриваемом классе задач не обеспечивает заданной погрешности и (или) не укладывается в установленные ресурсы по времени и памяти) [6—7].

Пожалуй, наиболее популярной труднорешаемой задачей комбинаторной оптимизации является известная

Задача коммивояжера. Найти перестановку  $\sigma = (i_k)$  ( $k = 1, \dots, n$ ) целых чисел  $1, \dots, n$ , минимизирующую длину замкнутого обхода  $n$  городов

$$\sum_{k=1}^n c_{i_k i_{k+1}} \rightarrow \min_{\sigma} \quad (1)$$

где  $(c_{ij})_{n \times n}$  — матрица расстояний между городами  $1 \leq i, j \leq n$ ; предполагается  $i_n = i_1$ .

Принадлежность ЗК к числу NP-трудных проблем [5] в определенной мере объясняет отсутствие эффективных точных методов ее решения. При этом ЗК является в некотором смысле типовой дискретной экстремальной задачей, на которой проверялись различные алгоритмические идеи в дискретной оптимизации. В частности, впервые именно к ЗК была применена известная схема метода ветвей и границ [12], в котором осуществляется отсечение целых подмножеств неперспективных вариантов. Однако даже такой сокращенный перебор с увеличением размерности задачи быстро выводит нас за пределы наших вычислительных возможностей.

В качестве одного из ранних примеров построения приближенного алгоритма с оценками для ЗК укажем на алгоритм, описанный в [13], где каждая задача получается в результате случайного выбора  $n$  точек в ограниченной, односвязной, с достаточно гладкой границей области  $m$ -мерного евклидова пространства,  $m \geq 2$ , при непрерывной всюду от-

личной от нуля функции плотности распределения точек. В случае равновероятного выбора точек в единичном квадрате для алгоритма из [13]  $T_{\mathcal{A}} = O(n \log_2 n)$ ,  $\Pi_{\mathcal{A}} = O(n)$ ,  $\epsilon_{\mathcal{A}} = 0,48$ ,  $\delta_{\mathcal{A}} \rightarrow 0$  при  $n \rightarrow \infty$ .

Особый интерес представляют малотрудоёмкие асимптотически точные (статистически эффективные) алгоритмы, в которых  $\epsilon_{\mathcal{A}}$ ,  $\delta_{\mathcal{A}}$  стремятся к нулю с ростом размерности задачи. Одним из первых результатов такого рода было обоснование условий статистической эффективности алгоритма «Иди в ближайший город, где еще не побывал»\*) для ЗК с классом  $\mathcal{A}$  матриц  $(c_{ij})_{n \times n}$ , элементы которых выбираются из отрезка  $[a_n, b_n]$ ,  $a_n > 0$ , вещественных чисел случайно и независимо друг от друга с одинаковой функцией распределения  $\mathcal{F}(x)$ ,  $a_n \leq x \leq b_n$  [9—10].

**Теорема 1.** В случае ЗК с матрицами  $(c_{ij})_{n \times n}$  из класса  $\mathcal{A}$  алгоритм «Иди в ближайший непройденный город» является асимптотически точным, если

$$\frac{b_n}{a_n} = o\left(\frac{n}{\max\{n\gamma_n; \mathcal{F}_n\}}\right), \quad (2)$$

$$\mathcal{F}_n = \int_{\gamma_n}^1 \frac{d\xi}{F_\xi} \rightarrow \infty \text{ при } n \rightarrow \infty, \quad (3)$$

где  $F_\xi$  — функция распределения случайной величины  $\xi = (x - a_n) / (b_n - a_n)$ ,  $0 \leq \xi \leq 1$ ;  $\gamma_n$  — корень уравнения  $F_\xi = n^{-1}$ . В случае равномерного распределения условия (2), (3) асимптотической точности этого алгоритма приводятся к виду  $b_n/a_n = o(n/\ln n)$ .

Для задачи коммивояжера на максимум длины обхода  $n$  городов соответствующие условия асимптотической точности выглядят несколько проще.

**Теорема 2.** В случае ЗК на максимум с матрицами  $(c_{ij})_{n \times n}$  из класса  $\mathcal{A}$  алгоритм «Иди в самый удаленный непройденный город» асимптотически точен, если  $\mathcal{F}_n = o(n)$ ,  $\mathcal{F}_n \rightarrow \infty$  при  $n \rightarrow \infty$ , где

$$\mathcal{F}_n = \int_0^{1-1/n} \frac{d\xi}{1-F_\xi}.$$

В частности, эти условия всегда выполняются в случае равномерного распределения  $\mathcal{F}(x) = (x - a_n) / (b_n - a_n)$ ,  $a_n \leq x \leq b_n$ .

Примечателен сам по себе факт, что в случае асимптотически точного алгоритма с увеличением размерности задачи мы почти всегда можем рассчитывать на получение все более точного решения.

## § 2. ЗАДАЧА КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ В УСЛОВИЯХ ОГРАНИЧЕННЫХ РЕСУРСОВ И ЗАДАННЫХ ДИРЕКТИВНЫХ СРОКОВ

Предполагаем, что анализируемый проект может быть представлен в виде совокупности операций (работ), взаимосвязанных между собой по технологии выполнения и по ресурсно-временным факторам и направленных на достижение единой цели [1—3]. Такой совокупности элементов и взаимосвязей проекта сопоставим детерминированную сетевую модель  $G(X, U)$  — ориентированный взвешенный граф (мультиграф) с выделенным полюсом  $\beta$ -концом проекта, где  $X$  — множество вершин сетевой модели, отождествляемых с событиями проекта,  $U \subset X \times X$  — множество дуг-работ проекта.

\*) Нетрудно заметить, что требуемые ресурсы по трудоёмкости и по памяти для алгоритма «Иди в ближайший...» ограничены величиной  $O(n^2)$ .

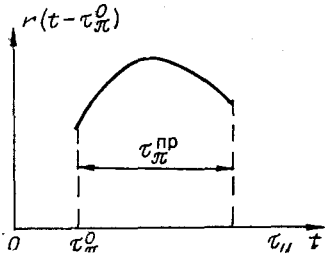


Рис. 1.

Через  $M$  обозначим множество типов ресурсов (продукции), задействованных в проекте. Пусть  $(U_1, U_2, U_3)$  — разбиение множества  $U$  соответственно на подмножества  $U_1$  (фактических работ),  $U_2$  (работ-ожиданий) и  $U_3$  (фиктивных работ). С потреблением и (или) производством ресурсов из множества  $M$  связаны только фактические работы. Для каждой работы  $u \in U$  задаются коды ее начального  $x_u$  и конечного  $y_u$  событий, а также (для всех работ, кроме фиктивных) ее продолжительность  $\tau_u$ . Кроме того, для каждой фактической работы  $u \in U_1$  задается следующая информация:

○ совокупность признаков, характеризующих данную работу в целом: приоритет, принадлежность к определенной отрасли, территориально-промышленному комплексу, подрядчику, зоне, региону и т. п.

○ множество  $\Pi_u$  профилей  $\pi$  работы  $u$ , заданных в виде

$$\{i_{\pi}, \tau_{\pi}^0, \tau_{\pi}^{\text{np}}, r_{\pi}(t) (0 \leq t \leq \tau_{\pi}^{\text{np}})\}, \pi \in \Pi_u, \quad (4)$$

где  $i_{\pi}$  — тип ресурса;

○  $\tau_{\pi}^0$  — сдвиг начала выполнения профиля  $\pi$  по отношению к моменту начала выполнения работы  $u$ ;

○  $\tau_{\pi}^{\text{np}}$  — длительность профиля  $\pi$ ;

○  $r_{\pi}(t) (0 \leq t \leq \tau_{\pi}^{\text{np}})$  — эпюра потребления ресурсов (выпуска продукции) профиля  $\pi$  в интервале  $[0, \tau_{\pi}^{\text{np}}]$ . Предполагается, что  $r_{\pi}(t) = 0$  при  $t \notin [0, \tau_{\pi}^{\text{np}}]$  (рис. 1).

Из множества  $M$  всех типов ресурсов, задействованных в проекте, выделено подмножество  $M^0 \subset M$  типов ограниченных потребляемых ресурсов. Для каждого ограниченного ресурса  $i \in M^0$  считается заданным количество  $B_i^t$  ресурсов, выделяемое в каждый год  $t = 1, \dots, T_i$ , где  $T_i$  — длительность интервала планирования с ограничением на ресурсы  $i$ -го типа. При  $t > T_i$  предполагается, что ограничение на ресурс  $i$ -го типа не накладывается.

Из множества вершин-событий сетевой модели выделено подмножество  $X_{\text{дир}} \subset X$  контрольных (директивных) событий, для каждого из которых задан директивный срок его свершения  $T_{\text{дир}}(x) > 0, x \in X_{\text{дир}}$ . Предполагается, что конец проекта  $\beta$  принадлежит  $X_{\text{дир}}$ . Все работы, входящие в директивное событие  $x \in X_{\text{дир}}$ , должны быть завершены не позже момента  $T_{\text{дир}}(x)$ .

Введем переменные величины  $t_u$  — времена начал выполнения работ  $u \in U$  сетевой модели. Совокупность  $\eta = (t_u) (u \in U)$  моментов начал выполнения работ сетевой модели назовем *допустимым календарным планом (расписанием)*, если выполняются следующие условия.

1°. Соблюдается технология выполнения работ:  $t_u + \tau_u \leq t_v$  для любой пары работ  $u, v \in U$  такой, что  $x_v = y_u$ .

2°. Не нарушаются директивные сроки:  $t_u + \tau_u \leq T_{\text{дир}}(x)$  для любой пары работы  $u \in U$  с  $y_u = x$ , где  $x \in X_{\text{дир}}$ .

3°. Имеющихся в наличии в каждый момент  $t = 1, \dots, T_i$  ограниченных ресурсов каждого типа  $i \in M^0$  хватает для всех работ, выполняемых в данный момент.

Формализуем запись условия 3°. Ресурс назовем *складируемым*, если он, будучи неистраченным в год  $t'$ , может быть использован в году  $t''$ ,  $t' < t''$ . Ресурс назовем *нескладируемым*, если он может быть использован только в год его выделения.

Обозначим

$M_c^0$  — множество типов ограниченных складируемых ресурсов;

$M_n^0$  — множество типов ограниченных нескладируемых ресурсов;

$b_t^i(\eta)$  — количество ограниченных ресурсов  $i$ -го типа,  $i \in M^0$ , задействованное в год  $t$  при расписании  $\eta$ ;

$\tilde{B}_t^i(\eta)$  — количество ограниченных ресурсов  $i$ -го типа,  $i \in M^0$ , имеющееся в наличии в год  $t$  при расписании  $\eta$ ;

$S_t^i(\eta) = \sum_{\tau=1}^t b_\tau^i(\eta)$  — суммарное количество ресурсов типа  $i$ , задействованное в годы  $1, \dots, t$ ;

$Q_t^i = \sum_{\tau=1}^t B_\tau^i$  — суммарное количество ограниченных ресурсов типа  $i \in M^0$ , выделенное в годы  $1, \dots, t$ .

Ясно, что

$$b_t^i(\eta) = \sum_{u \in U_1} \sum_{\pi \in \Pi_u^i} \int_{t-1}^t r_\pi(t' - t_u - \tau_\pi^0) dt',$$

где  $\Pi_u^i = \{\pi \mid i_\pi = i, \pi \in \Pi_u\}$ ,  $i \in M_x$ ,  $u \in U_1$ .

Для нескладируемых ограниченных ресурсов типа  $i \in M_n^0$  величина имеющихся в наличии ресурсов не зависит от выбора календарного плана, так что  $\tilde{B}_t^i(\eta) = B_t^i$  для всякого  $t = 1, \dots, T_i$ . В случае же складируемых ресурсов типа  $i \in M_c^0$  его наличие в год  $t$  определяется суммой выделенных ресурсов  $B_t^i$  и остатков ресурсов этого типа, не использованных в предшествующие годы для данного расписания:  $\tilde{B}_t^i(\eta) = B_t^i + (Q_{t-1}^i - S_{t-1}^i(\eta))$ ,  $t = 1, \dots, T_i$ ,  $i \in M_c^0$ . Теперь условие 3<sup>о</sup> достаточности ограниченных ресурсов можно записать в виде

$$b_t^i(\eta) \leq \tilde{B}_t^i(\eta) \text{ для всяких } t = 1, \dots, T_i, i \in M_c^0.$$

Сформулируем основную задачу календарного планирования.

Задача 1. Среди всех допустимых расписаний  $\eta = (t_u) (u \in U)$  найти расписание минимальной длительности:

$$L(\eta) = \max_{u \in U} (t_u + \tau_u) \rightarrow \min_{\eta} \quad (5)$$

Длину оптимального расписания согласно критерию  $L(\eta)$  обозначим  $L^* = L(\eta^*)$ .

Наряду с основным критерием (5) будем также рассматривать дополнительный критерий минимизации суммы отклонений потребляемых ресурсов от заданных ограничений вида

$$\sum_{i \in M^0} \sum_{t=1}^{\min\{L^*, T_i\}} |B_t^i - b_t^i(\eta)| \rightarrow \min_{\eta \in (\eta^*)} \quad (6)$$

Ясно, что если существует допустимое расписание  $\eta$  минимальной длительности с  $b_t^i(\eta) = B_t^i$  ( $t = 1, \dots, T_i$ ,  $i \in M^0$ ), то приведенный выше критерий достигает минимально возможного нулевого значения.

Задача 1 в общем случае труднорешаема. В частности, в [14] для нее доказана NP-полнота проблемы существования допустимого расписания.

Если ограничений на ресурсы нет, то задача решается с линейной относительно числа работ сложностью, а именно: в качестве оптимального расписания можно взять, например, совокупность наиболее ранних моментов начал работ:  $\eta^1 = (t_u^1) (u \in U)$ . При этом, как известно, весь проект завершается за так называемое критическое время  $T_{кр}$ , равное длине максимального пути в сетевой модели.

$L$ -поздним расписанием назовем такое расписание  $\eta(L) = (t_u^{n,L})$  ( $u \in U$ ), для которого при выполнении 1°, 2° соблюдается следующее условие.

4°. Для всякого  $u \in U$  справедливо неравенство  $t_u^{n,L} + \tau_u \leq L$ , и увеличение любого  $t_u^{n,L}$  приводит к нарушению хотя бы одного из условий 1°, 2°, 4°.

Полагаем, что известны ранги  $\rho(x)$ ,  $x \in X$ , вершин сетевой модели. (Рангом  $\rho(x)$  вершины  $x \in X$  называют максимальное число дуг в пути из начала сетевой модели в данную вершину.) Справедлива

**Лемма 1.**  $L$ -позднее расписание можно найти с трудоемкостью, линейно зависящей от числа работ сетевой модели.

Доказательство следует из того, что  $L$ -позднее расписание может быть найдено по совокупности наиболее поздних моментов свершения событий  $(T_x^n) (x \in X)$ , вычисленных при замене директивных сроков величинами  $\min\{L; T_{\text{дир}}(x)\}$ ,  $x \in X_{\text{дир}}$ . Это осуществляется с трудоемкостью, линейно зависящей от числа работ.

Действительно, наиболее поздние моменты свершения событий  $T_x^n$ ,  $x \in X$ , как известно, можно вычислить за один просмотр списка  $U$  работ по формулам

$$T_{x_u}^n = \min\{T_{x_u}^n; T_{y_u}^n - \tau_u\}, \quad u \in U,$$

(предварительно упорядочив работы по убыванию рангов их конечных вершин-событий и положив  $T_x^n = L$ ,  $x \in X \setminus X_{\text{дир}}$ ). Затем вычисляем наиболее поздние моменты начал работ:  $t_u^{n,L} = T_{y_u}^n - \tau_u$ ,  $u \in U$ .

**Следствие 1.** В случае неограниченных ресурсов  $T_{\text{кр}}$ -позднее расписание  $\eta(T_{\text{кр}})$  в задаче 1 оптимально.

### § 3. СЛУЧАЙ СКЛАДИРУЕМОСТИ ОГРАНИЧЕННЫХ РЕСУРСОВ В ЗАДАЧЕ 1

При корректно заданных исходных данных  $T_{\text{кр}} \leq L^* \leq T_{\text{кр}} + T_{\text{max}}$ , где  $T_{\text{max}} = \max_{i \in M^0} T_i$ .

**Лемма 2.** Если ограниченные ресурсы складирuemы, то  $L^*$ -позднее расписание  $\eta(L^*)$  оптимально.

Доказательство. Расписание  $\eta(L^*)$  удовлетворяет условиям 1°, 2° по построению. Покажем, что оно удовлетворяет и 3°. Для расписаний  $\eta^* = (t_u^*) (u \in U)$  и  $\eta(L^*) = (t_u^n) (u \in U)$  имеем  $t_u^* \leq t_u^n$ ,  $u \in U$ . Пусть  $r_\pi(t - t_u^* - \tau_\pi^0)$  — интенсивность использования ресурсов типа  $i_\pi$  профиля  $\pi \in \Pi_u$ ,  $u \in U$ , в некоторый момент  $t \in (t_u^* + \tau_\pi^0, t_u^* + \tau_\pi^0 + \tau_\pi^{\text{пр}}]$  при оптимальном расписании  $\eta^*$ . Тогда в силу складирuemости ограниченных ресурсов из  $t_u^* \leq t_u^n$ ,  $u \in U$ , следует возможность использования тех же ресурсов для указанного профиля при  $L^*$ -позднем расписании, но в момент, сдвинутый вправо по временной оси на величину  $\Delta t = t_u^n - t_u^* \geq 0$ . Следовательно, расписание  $\eta(L^*)$ , как и расписание  $\eta^*$ , удовлетворяет условию 3°.

Кроме того, из условий  $t_u^* \leq t_u^n$ ,  $u \in U$ , имеем

$$L^* = \max_{u \in U} (t_u^* + \tau_u) \leq \max_{u \in U} (t_u^n + \tau_u) \leq L^*$$

и, таким образом,  $\max_{u \in U} (t_u^n + \tau_u) = L^*$ , т. е.  $\eta(L^*)$  — оптимальное расписание.

**Теорема 3.** Если ограниченные ресурсы складирuemы, то для задачи 1 можно построить асимптотически точное расписание за число действий  $s \log_2 n$ , где  $s$  не зависит от числа работ  $n$ .

Доказательство. Ввиду леммы 2 расписание  $\eta(L)$  существует тогда и только тогда, когда  $L \geq L^*$ . Задавшись величиной  $\varepsilon$  — относительной погрешности в определении длины расписания — мы с помощью дихотомического процесса по значениям  $L$ ,  $T_{кр} \leq L \leq T_{кр} + T_{max}$ , можем найти расписание  $\eta(L_\varepsilon^*)$ , где  $L_\varepsilon^* \leq (1 + \varepsilon)L^*$ , за число итераций  $\sim \log_2(T_{max}/(\varepsilon L^*))$ . По лемме 1 одна итерация требует числа действий, линейного по числу работ. Таким образом, расписание  $\eta(L_\varepsilon^*)$  можно найти с трудоемкостью порядка  $n \log_2(T_{max}/(\varepsilon L^*))$ . Положим  $\varepsilon = T_{max}/(nT_{кр})$ . Тогда трудоемкость алгоритма построения  $\eta(L_\varepsilon^*)$  ограничена величиной порядка  $n \log_2 n$  и  $\varepsilon \rightarrow 0$  с ростом числа работ  $n$ , т. е. алгоритм является асимптотически точным.

#### § 4. АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ 1 С ДОПОЛНИТЕЛЬНЫМ КРИТЕРИЕМ (6) В СЛУЧАЕ СКЛАДИРУЕМОСТИ ОГРАНИЧЕННЫХ РЕСУРСОВ

Опишем приближенный алгоритм, состоящий из двух этапов. На первом строится асимптотически точное относительно критерия (5) решение задачи 1, т. е.  $L_\varepsilon^*$ -позднее расписание  $\eta(L_\varepsilon^*)$  за  $cn \log_2 n$  действий, как это описано выше. На втором этапе осуществляется последовательный сдвиг работ сетевой модели влево по временной оси.

Работу  $u \in U$  предполагаем наложенной на календарь, если окончательно установлен момент ее начала. В начале второго этапа все работы считаются неналоженными на календарь. Наложение любой работы на календарь может осуществляться только при условии наложения на календарь всех предшествующих ей работ. Множество неналоженных на календарь работ, для которых все предшествующие работы на календарь наложены, будем называть *текущим фронтом работ*. Выбираем очередную работу из текущего фронта по весовой функции, учитывающей такие факторы, как приоритет работы, объем потребляемых ресурсов, величину резервного интервала, длительность работы, наиболее ранний момент начала этой работы без учета условия 3° (после наложения на календарь предшествующих работ) и т. п. Таким образом, вычисление весовой функции для некоторой работы предполагается возможным, вообще говоря, только после наложения всех предшествующих ей работ. Трудоемкость вычисления весовой функции ограничим величиной  $O(\log_2 n)$ . В начале этапа в текущий фронт включаются все работы, выходящие из вершин нулевого ранга. Далее осуществляется  $n$  шагов второго этапа. На шаге  $k$ ,  $1 \leq k \leq n$ , из текущего фронта выбирается работа, имеющая максимальное значение весовой функции. После этого выбранная работа сдвигается в пределах резервного интервала на возможно меньший момент времени с учетом выполнения ресурсных ограничений 3° для всех работ сетевой модели (как уже наложенных, так и неналоженных на календарь). Установление момента начала выбранной работы будем осуществлять с относительной точностью  $\varepsilon = O(1/n)$ , стремящейся к нулю с ростом числа работ.

После наложения на календарь работа  $u = (x, y)$  исключается из текущего фронта работ, и если все работы, предшествующие вершине  $y$ , наложены на календарь, то в текущий фронт включаются работы, выходящие из вершины  $y$ . При этом одновременно вычисляются значения весовой функции каждой из этих работ. Если текущий фронт работ не пуст, переходим на следующий шаг второго этапа.

**Теорема 4.** *Описанный двухэтапный алгоритм можно осуществить за  $cn \log_2 n$  действий.*

Доказательство. Ввиду приведенных рассуждений осталось оценить трудоемкость второго этапа, состоящего из  $n$  шагов. На каждом шаге с точностью до  $\varepsilon$  должен быть установлен возможно меньший

момент начала работы, накладываемой на календарь. Методом дихотомии это можно сделать за число действий порядка  $\log_2(1/\varepsilon)$ , а ввиду  $\varepsilon = O(1/n)$  — порядка  $\log_2 n$ . Таким образом, число действий, затрачиваемых на наложение на календарь всех работ, ограничено величиной  $O(n \log_2 n)$ .

Количество действий на вычисление значений весовых функций работ также не превышает величины  $O(n \log_2 n)$ . Покажем теперь, что организация последовательного выбора из текущего фронта очередной работы с максимальным значением весовой функции и соответствующая корректировка нового текущего фронта может быть осуществлена за число действий, ограниченное величиной  $O(n \log_2 n)$ .

Будем хранить множество из  $N$  работ текущего фронта в виде списковой структуры, известной под названием *двоичного дерева* (ДД). Каждой вершине  $j = 1, \dots, N$  этого дерева сопоставлена работа текущего фронта. В каждую вершину  $j$ ,  $j \neq 1$ , входит одно ребро, из каждой вершины выходит не более двух ребер. При  $j > N/2$  вершина  $j$  висячая, при  $j < N/2$  из  $j$  выходит два ребра в вершины  $2j$  и  $2j + 1$ ; при  $j = N/2$  из  $j$  выходит только одно ребро в вершину  $N$ .

Двоичное дерево назовем *пирамидальным*, если вдоль любого пути от корневой вершины ДД к любой висячей вершине значения весов работ, сопоставленных вершинам, не возрастают. Таким образом, в корневой вершине пирамидального двоичного дерева (ПДД) хранится работа с максимальным значением весовой функции. Непирамидальное ДД назовем *полупирамидальным*, если его поддеревья с корневыми вершинами 2 и 3 являются пирамидальными.

Следующие две леммы очевидны.

**Лемма 3.** *Полупирамидальное  $N$ -вершинное ДД можно доупорядочить в ПДД за число действий, ограниченное величиной  $O(\log_2 n)$ .*

**Лемма 4.**  *$N$ -вершинное ПДД при добавлении к нему  $(N + 1)$ -го элемента (вершины) может быть доупорядочено в  $(N + 1)$ -вершинное ПДД за число действий, ограниченное величиной  $O(\log_2 n)$ .*

Нетрудно видеть, что если текущий фронт работ представлен в виде ПДД, то выбор работы с максимальным значением весовой функции осуществляется за число действий, ограниченное константой.

Покажем теперь, что все корректировки текущего фронта в ходе всего второго этапа можно осуществить за время  $O(n \log_2 n)$ . Действительно, после наложения очередной работы на календарь удаляем ее из текущего фронта, перемещаем  $N$ -ю вершину фронта в корневую вершину, уменьшаем  $N$  на единицу и по лемме 3 доупорядочиваем полученное полупирамидальное двоичное дерево в ПДД за  $O(\log_2 n)$  действий. Далее проверяем, не появились ли новые работы для ввода их в текущий фронт. Если появились, то вводим последовательно каждую такую работу согласно лемме 4 за  $O(\log_2 n)$  действий, увеличивая при этом  $N$  на единицу.

Операций наложения работ на календарь и ввода новой работы в текущий фронт происходит не более  $n$  раз, откуда общая трудоемкость корректировки текущего фронта ограничена величиной  $O(n \log_2 N_{\text{фр}})$ , где  $N_{\text{фр}}$  — максимальное число работ текущего фронта. Поскольку  $N_{\text{фр}} \leq n$ , то и общая трудоемкость второго этапа алгоритма ограничена величиной, зависящей от числа работ сетевой модели как функция  $cn \log_2 n$ , где  $c$  — константа, не зависящая от  $n$ .

## § 5. СЛУЧАЙ НЕСКЛАДИРУЕМЫХ ОГРАНИЧЕННЫХ РЕСУРСОВ

Опишем малотрудоемкий алгоритм с апостериорной оценкой точности получаемой длины расписания в случае, когда имеют место ограничения на ресурсы обоих типов. Как в предыдущем случае, работа алгоритма осуществляется в два этапа.

На первом в предположении складированности всех ограниченных



ресурсов за  $cn \log_2 n$  действий отыскивается оценочное расписание  $(\tilde{t}_u)$  ( $u \in U$ ), асимптотически точное по основному критерию (5).

На втором этапе ищется допустимое расписание с учетом дополнительного критерия (6). С использованием весовой функции формируется текущий фронт работ, представляемый с помощью ПДД. Выбираемые из текущего фронта работы последовательно накладываются на календарь. После наложения на календарь  $k$  работ,  $0 \leq k < n$ , ищется возможно меньший момент  $t_u$  начала  $(k+1)$ -й по порядку работы, причем пока  $t_u \leq \tilde{t}_u$  для уже наложенных на календарь работ, начала всех работ, кроме  $(k+1)$ -й, считаются закрепленными и равными  $t_u$  для всех наложенных работ и  $\tilde{t}_u$  для оставшихся работ. Как только для очередной накладываемой работы момента  $t_u \leq \tilde{t}_u$  найти не удалось, закрепленными считаем лишь моменты начал работ, уже наложенных на календарь.

В результате работы алгоритма может оказаться, что допустимое решение не найдено либо будет указана апостериорная оценка точности полученного решения. Действительно, пусть  $L_{\mathcal{A}}$  — длина расписания, полученного в результате работы алгоритма,  $\tilde{L}^*$  и  $\tilde{L}$  — длины соответственно оптимального и оценочного расписаний в предположении складированности ограниченных ресурсов:  $\tilde{L} \leq (1 + \varepsilon) \tilde{L}^*$ ,  $\varepsilon = O(1/n)$ . Относительная погрешность  $\varepsilon_{\mathcal{A}}$  определения длительности расписания может быть оценена следующим образом:

$$\varepsilon_{\mathcal{A}} = \frac{L_{\mathcal{A}} - L^*}{L^*} \leq \frac{L_{\mathcal{A}}}{L^*} - 1 \leq (1 + \varepsilon) L_{\mathcal{A}} / \tilde{L} - 1.$$

Трудоемкость реализации алгоритма в зависимости от числа работ сетевой модели также ограничена величиной  $cn \log_2 n$ .

Далее мы рассмотрим более простую задачу, сводимую к задаче 1.

## § 6. ЗАДАЧА УПАКОВКИ В КОНТЕЙНЕРЫ В ОДНОМЕРНОМ СЛУЧАЕ

Эта задача [15] заключается в размещении предметов из списка  $S = (a_i)$  ( $i = 1, \dots, n$ ) в минимальное количество контейнеров с заданной вместимостью  $B$ ,  $B \geq a_i > 0$ ,  $i = 1, \dots, n$ . Задача 1 эквивалентна задаче упаковки в контейнеры в случае независимости работ, постоянных длительностях работ и постоянной величине ограниченного ресурса  $B$  одного типа. При этом минимизация времени выполнения совокупности работ равносильна минимизации числа задействованных контейнеров.

Несмотря на значительное упрощение постановки задачи, мы тем не менее остаемся в классе NP-трудных проблем [5]. Большинство работ по задаче упаковки в контейнеры посвящены разработке приближенных алгоритмов с гарантированными оценками максимального отклонения получаемого решения от точного. К наилучшим таким алгоритмам принадлежит алгоритм FFD («первый пригодный с упорядочением») с оценкой относительного отклонения  $\varepsilon = 2/9$  [15]. Ниже опишем асимптотически точный подход к решению одномерной задачи упаковки в контейнеры при  $a_i \in \{1, \dots, B\}$  и определенных предположениях на характеристическую функцию  $\chi_S$  списка  $S = (a_i)$  ( $i = 1, \dots, n$ ):  $\chi_S(k) = v_k$ ,  $v_k$  — число предметов веса  $k$  в списке  $S$ ,  $k = 1, \dots, B$ .

Функцию  $f$  целочисленного аргумента  $k = 1, \dots, B$  назовем *b-симметричной*, если  $f(k) = f(b - k)$  при  $k = 1, \dots, [b/2]$  и  $f(k) = 0$  при  $k \geq b$ . Функцию  $f$  при  $B = 2^q$ ,  $Q > 0$  — целое, будем называть *B-регулярной*, если она представима в виде конечной суммы  $2^q$ -симметричных функций  $f_q$ ,  $q = 1, \dots, Q$ :  $f(k) = \sum_{q=1}^Q f_q(k)$ . Свойством, которым обладает

характеристическая функция списка, будем обозначать и сам список. Например, в случае  $b$ -симметричной характеристической функции список будем называть  $b$ -симметричным.

В работе [11] показано, что класс  $B$ -регулярных списков включает в себя списки с невозрастающими характеристическими функциями  $\chi_s(k)$ ,  $k = 1, \dots, B$ , при  $B = 2^Q$ . Там же описан следующий алгоритм  $\mathcal{A}_1$  упаковки. Сначала объединяются попарно предметы с суммарным весом  $B$ . Каждая такая пара целиком заполняет один контейнер. Затем действуем поэтапно: на этапе с номером  $q$ ,  $1 \leq q < Q - 1$ , составляем пары из оставшихся предметов с суммарным весом  $B/2^q$ , после чего в каждый контейнер укладываем по  $2^q$  таких пар. Предметы, которые ни на одном из этапов не были уложены в контейнер, упаковываем с помощью какого-либо алгоритма, имеющего линейную трудоемкость (например, NF-«следующий пригодный» [15]).

**Лемма 5.** В случае детерминированных  $B$ -симметричных или  $B$ -регулярных списков алгоритм  $\mathcal{A}_1$  приводит к точному решению задачи упаковки в контейнеры. При  $B \leq cn$ , где  $c$  — положительная константа, алгоритм  $\mathcal{A}_1$  имеет трудоемкость, линейную относительно  $n$ . Число контейнеров, необходимых для упаковки предметов из списков указанного

вида, равно  $L_{\mathcal{A}_1} = L^* = \lceil L^0 \rceil$ , где  $L^0 = \frac{1}{B} \sum_{k=1}^B k v_k$ ,  $v_k = \chi_s(k)$ ,  $k = 1, \dots, B$ ,  $\lceil x \rceil = \min \{i \mid i \geq x, i \text{ целое}\}$ .

Нетрудно видеть, что алгоритм  $\mathcal{A}_1$  приводит к оптимальной упаковке детерминированного списка, состоящего из  $B$ -симметричного подсписка и подсписка предметов, попарно не помещающихся в один контейнер. Отсюда следует, что алгоритм  $\mathcal{A}_1$  приводит к оптимальному решению также в случае детерминированных неубывающих списков. Заметим, что добавление к списку  $S$  предметов веса 1 или  $B$  может только улучшить оценку качества алгоритма.

Итак, мы описали несколько классов задач упаковки в контейнеры с детерминированным распределением весов предметов, при которых оказывается возможным построение малотрудоемкого точного алгоритма. Далее рассмотрим класс  $\mathcal{X}_n$ , содержащий списки  $S = (a_i)$  ( $i = 1, \dots, n$ ) весов предметов, формируемых с помощью схемы  $n$  последовательных независимых испытаний, при каждом из которых очередной предмет  $a_i$  попадает в один из весовых классов  $k = 1, \dots, B$  в соответствии с одинаковой дискретной функцией распределения

$$p_k = P\{a_i = k\} \geq 0, \quad k = 1, \dots, B; \quad \sum_{k=1}^B p_k = 1.$$

Соответствующее свойство функции распределения для конкретной реализации случайного списка, вообще говоря, не сохраняется. Поэтому далее мы займемся вероятностным анализом модифицированного приближенного алгоритма  $\tilde{\mathcal{A}}_1$  решения задач из класса  $\mathcal{X}_n$ . Модифицированный алгоритм использует решение полученное с помощью алгоритма  $\mathcal{A}_1$ , для некоторого измененного по сравнению с исходным списка  $\tilde{S}$ .

Ниже нам понадобится неравенство Бернштейна для вероятности отклонений суммы  $v = \sum_{j=1}^n \xi_j$  независимых случайных величин  $\xi_1, \dots, \xi_n$  с одинаковым математическим ожиданием  $a$  и дисперсией  $\beta$ .

**Лемма 6.** [16]. При  $|M(\xi_j - a)^m| \leq (\beta/2) H^{m-2} m!$  ( $m \geq 2$ ), где  $H$  — положительная константа, выполнено неравенство  $P\{|v - na| \geq 2t\sqrt{n\beta}\} \leq 2^{-t^2}$ , если только  $0 < t \leq \sqrt{n\beta}/2H$ .

**Следствие 2.** Для схемы Бернулли  $P\{\xi_j = 1\} = p$ ,  $P\{\xi_j = 0\} = 1 - p$ ,  $j = 1, \dots, n$ ,  $\beta = p(1-p)$  при  $(\ln n)/n \leq p \leq 1 - (\ln n)/n$  имеем  $P\{|v - np| \geq 2\sqrt{n\beta \ln n}\} \leq 2/n$ .

Доказательство. Положив  $H = 1/3$  и  $t = \sqrt{\ln n}$ , убеждаемся в соблюдении условий леммы 6 в случае  $(\ln n)/n \leq p \leq 1 - (\ln n)/n$ :

$$|M(\xi_j - p)^m| \leq M(\xi_j - p)^2 = \beta \leq \frac{2 \cdot 3 \cdot \dots \cdot m}{2 \cdot 3 \cdot \dots \cdot 3} \beta = \frac{\beta}{2} H^{m-2} m! \quad (m \geq 2),$$

$$\frac{\sqrt{n\beta}}{2H} = \frac{3}{2} \sqrt{np(1-p)} \geq \frac{3}{2} \min_p \sqrt{np(1-p)} =$$

$$= \frac{3}{2} \sqrt{\ln n (1 - \ln n/n)} > \sqrt{\ln n} = t.$$

**Теорема 5.** Алгоритм  $\tilde{\mathcal{A}}_1$  на классе задач  $\mathcal{K}_n$  с  $B$ -симметричным распределением  $p(k)$ ,  $k = 1, \dots, B$ , асимптотически точен, если

$$B = o(n/\ln n). \quad (7)$$

Доказательство. Введем обозначения

$$I_1 = \left\{ k \mid \frac{\ln n}{n} \leq p_k \leq 1 - \frac{\ln n}{n}, \quad k = 1, \dots, B \right\},$$

$$I_0 = \left\{ k \mid 0 < p_k < \frac{\ln n}{n}, \quad k = 1, \dots, B \right\},$$

$$v_0 = \sum_{k \in I_0} v_k, \quad p_0 = \sum_{k \in I_0} p_k,$$

$$\Delta_0 = n\sqrt{p_0(1-p_0)}/2 \quad \text{при } p_0 < (\ln n)/n,$$

$$\Delta_k = 2\sqrt{np_k(1-p_k)\ln n}, \quad k \in I_1 \cup \{0\}, \quad p_k \geq (\ln n)/n,$$

$$A_k - \text{событие } \{|v_k - np_k| \geq \Delta_k\}, \quad k \in I_1 \cup \{0\}.$$

Пусть  $\{S'\}$  — множество списков из  $\mathcal{K}_n$ , в которых не выполнено ни одно из событий  $A_0, A_k$ ,  $k \in I_1$ . Оценим вероятность  $\delta_n$  того, что случайный список  $S$  не содержится в  $\{S'\}$ :

$$\delta_n = P\left(A_0 + \bigcup_{k \in I_1} A_k\right) \leq P(A_0) + \sum_{k \in I_1} P(A_k).$$

По следствию 2  $P(A_k) \leq 2/n$  при  $k \in I_1$  и  $k = 0$ ,  $p_0 \geq (\ln n)/n$ , а при  $p_0 < (\ln n)/n$  по неравенству Чебышева имеем

$$P(A_0) \leq \frac{np_0(1-p_0)}{(n\sqrt{p_0(1-p_0)}/2)^2} = \frac{2}{n},$$

откуда  $\delta_n \leq 2(1 + |I_1|)/n = 2B/n \rightarrow 0$  при  $n \rightarrow \infty$ .

Далее будем рассматривать списки из множества  $\{S'\}$ , имея в виду, что произвольный список  $S$  содержится в нем с вероятностью, стремящейся к 1 при  $n \rightarrow \infty$ . Решение строим следующим образом. Образует список  $\tilde{S}$  с характеристической функцией

$$\chi_{\tilde{S}}(k) = \begin{cases} [np_k + \Delta_k] & \text{при } k \in I_1, \\ 0 & \text{при } k \in I_0; \end{cases}$$

$v_0$  предметов с весами  $k \in I_0$  помещаем по одному в каждый контейнер, а предметы из детерминированного списка  $\tilde{S}$  в силу его  $B$ -симметричности укладываем алгоритмом  $\mathcal{A}_1$  в

$$\tilde{L} = \left[ \frac{1}{B} \sum_{k \in I_1} k [np_k + \Delta_k] + 1 \right]$$

контейнеров согласно лемме 5. Удаляя из контейнеров предметы, не принадлежащие исходной реализации случайного списка, получим решение с числом контейнеров  $L_{\tilde{\mathcal{A}}_1} \leq \tilde{L} + v_0 \leq \tilde{L} + np_0 + \Delta_0 = L_{\text{верх}}$ .

С другой стороны,

$$L_{\tilde{\mathcal{A}}_1} \geq L^* \geq L^0 \geq \frac{1}{B} \sum_{k \in I_1} k(n p_k - \Delta_k) = L_{\text{ниж}}.$$

Оценим погрешность получаемого решения с учетом тождества  $\sum_{k=1}^B k f(k) = \frac{B}{2} \sum_{k=1}^B f(k)$ , справедливого для  $B$ -симметричной функции  $f$ :

$$\begin{aligned} \varepsilon_n &= \frac{L_{\tilde{\mathcal{A}}_1} - L^*}{L^*} \leq \frac{L_{\text{верх}} - L_{\text{ниж}}}{L_{\text{ниж}}} \leq \\ &\leq \frac{\sum_{k \in I_1} k(n p_k + \Delta_k) + B(n p_0 + \Delta_0) - \sum_{k \in I_1} k(n p_k - \Delta_k) + B}{\sum_{k \in I_1} k(n p_k - \Delta_k)} \leq \\ &\leq \frac{\sum_{k \in I_1} \Delta_k + \Delta_0 + n p_0 + 1}{0,5 \left( n(1 - p_0) - \sum_{k \in I_1} \Delta_k \right)}. \end{aligned}$$

Ввиду неравенств

$$\begin{aligned} p_0 &= \sum_{i \in I_0} p_k < |I_0|(\ln n)/n, \\ \sum_{k \in I_1} \Delta_k &\leq 2 \sqrt{n \ln n} \sum_{k \in I_1} \sqrt{p_k} \leq 2 \sqrt{B n \ln n}, \\ \Delta_0 &\leq \sqrt{n \ln n} \end{aligned}$$

и обозначения  $\alpha_n = \sqrt{B(\ln n)}/n$  имеем

$$\varepsilon_n \leq \frac{(1 + 2\sqrt{B})\sqrt{n \ln n} + B \ln n + 1}{0,5(n - B \ln n - 2\sqrt{B n \ln n})} < \frac{6\alpha_n(1 + \alpha_n)}{1 - 2\alpha_n - \alpha_n^2} \sim 6\alpha_n,$$

откуда в силу (7)  $\varepsilon_n \rightarrow \infty$  при  $n \rightarrow \infty$ . Теорема доказана.

**Следствие 3.** Алгоритм  $\tilde{\mathcal{A}}_1$  на классе задач  $\mathcal{K}_n$  с равномерным распределением весов предметов  $k = 1, \dots, B$  асимптотически точен при  $B = o(n/\ln n)$ .

**Теорема 6.** Алгоритм  $\tilde{\mathcal{A}}_1$  на классе  $\mathcal{K}_n$  задач упаковки в контейнеры с невозрастающей функцией распределения  $p_k, k = 1, \dots, B$  (где  $B = 2^Q, Q$  — целое), асимптотически точен, если

$$\sum_{k=1}^B k \sqrt{p_k(1-p_k)} \Big/ \sum_{k=1}^B k p_k = o(\sqrt{n/\ln n}).$$

Доказательство строится по той же схеме, что и в теореме 5. При этом учитываем, что из невозрастания функции  $p_k, k = 1, \dots, B$ , следует невозрастание характеристической функции  $\chi_{\tilde{\mathcal{S}}}(k)$  списка  $\tilde{\mathcal{S}}, \chi_{\tilde{\mathcal{S}}}(k) = [n p_k + \Delta_k], k = 1, \dots, B$ , откуда получаем  $B$ -регулярность списка  $\tilde{\mathcal{S}}$ .

Аналогичным способом можно получить следующее утверждение, касающееся оценки отклонения числа контейнеров  $L_{\tilde{\mathcal{A}}_1}$ , получаемого в результате работы алгоритма  $\tilde{\mathcal{A}}_1$ , от тривиальной нижней оценки  $L^0$  требуемого числа контейнеров:

**Теорема 7.** Для задач из класса  $\mathcal{K}_n$  с функцией распределения  $p_k, k = 1, \dots, B$ , обладающей свойством  $B$ -симметричности или  $B$ -регулярности, найдутся такие  $\varepsilon_n \rightarrow 0, \delta_n \rightarrow 0$  при  $n \rightarrow \infty$ , что

$$\mathbf{P}\{L_{\tilde{\mathcal{A}}_1} \geq (1 + \varepsilon_n)L^0\} \leq \delta_n. \quad (8)$$

Будем называть *квазирегулярной* функцию распределения  $p_k$ ,  $k = 1, \dots, B$ , весов предметов случайного списка  $S = (a_i)$  ( $i = 1, \dots, n$ ), при которой алгоритм  $\mathcal{A}_1$  имеет отклонение  $L_{\mathcal{A}_1}$  от  $L^0$  с оценками  $\varepsilon_n \rightarrow 0$ ,  $\delta_n \rightarrow 0$  при  $n \rightarrow \infty$ . Из сказанного выше следует, что равномерное  $B$ -симметричное и  $B$ -регулярное распределения являются квазирегулярными.

## § 7. ДВУМЕРНАЯ ЗАДАЧА УПАКОВКИ В ПОЛОСУ

Рассмотрим теперь двумерную задачу упаковки в полосу высотой  $B$  предметов списка  $S = (a_i, \tau_i)$  ( $i = 1, \dots, n$ ), каждый из которых помимо своего веса  $a_i$  характеризуется длительностью  $\tau_i$ ,  $\tau_i \in \{1, \dots, T\}$ ,  $T$  — целочисленная константа. Требуется упаковать предметы списка  $S$  в полосу минимальной длины. Если  $\tau_i$  — константа,  $i = 1, \dots, n$ , то эта задача эквивалентна задаче упаковки в контейнеры.

Обозначим  $S_\tau = \{(a_i, \tau_i) | \tau_i = \tau, i = 1, \dots, n\}$  — подмножество списка  $S$  с элементами длительности  $\tau_i = \tau$ ,  $\tau = 1, \dots, T$ ;  $W(S') = \sum_{i|(a_i, \tau_i) \in S'} a_i \tau_i$  для любого  $S' \subset S$ ;

$p(k, \tau)$  — функция распределения весов и длительностей предметов,  $p(k, \tau) \geq 0$ ,  $k = 1, \dots, B$ ,  $\tau = 1, \dots, T$ ;

$$\sum_k \sum_\tau p(k, \tau) = 1;$$

$\mathcal{A}_2$  — алгоритм, последовательно применяющий алгоритм  $\mathcal{A}_1$  решению одномерной задачи упаковки в контейнеры к спискам  $S_\tau$ ,  $\tau = 1, \dots, T$ ;

$$A(S') = \sum_{i|(a_i, \tau_i) \in S'} a_i, \quad S' \subset S.$$

Будем говорить, что функция  $p(k, \tau)$  *квазирегулярна*, если для любого  $\tau = 1, \dots, T$  квазирегулярна функция

$$p_k = p(k, \tau) / \varphi_k, \quad \varphi_k = \sum_{\tau=1}^T p(k, \tau), \quad k = 1, \dots, B.$$

**Теорема 8.** Если  $p(k, \tau)$  квазирегулярна, то алгоритм  $\mathcal{A}_2$  приводит к асимптотически точному решению двумерной задачи упаковки в полосу.

*Доказательство.* Обозначим  $\varepsilon_\tau = L_\tau / (A(S_\tau) / B) - 1$ ,  $\tau = 1, \dots, T$ , где  $L_\tau$  — число контейнеров, полученное в результате применения алгоритма  $\mathcal{A}_1$  к списку  $S_\tau$ . Имеем  $L_\tau = (1 + \varepsilon_\tau) (A(S_\tau) / B)$ ,  $\tau = 1, \dots, T$ . Длина полосы, полученная в результате алгоритма  $\mathcal{A}_2$ , равна

$L_{\mathcal{A}_2} = \sum_{\tau=1}^T \tau L_\tau$ . Оценим относительную погрешность полученного решения. С учетом оценки снизу  $L^* \geq W(S) / B$  для целевой функции в задаче двумерной упаковки в полосу получаем

$$\begin{aligned} \varepsilon_{\mathcal{A}_2} &= \frac{L_{\mathcal{A}_2}}{L^*} - 1 \leq \frac{\sum_{\tau=1}^T \tau L_\tau}{W(S) / B} - 1 = \frac{\sum_{\tau=1}^T (1 + \varepsilon_\tau) \tau A(S_\tau) / B}{W(S) / B} - 1 \leq \\ &\leq \frac{\sum_{\tau=1}^T (1 + \varepsilon_\tau) W(S_\tau)}{W(S)} - 1 \leq (1 + \varepsilon_{\max}) \sum_{\tau=1}^T W(S_\tau) / W(S) - 1 = \varepsilon_{\max}, \end{aligned}$$

где  $\varepsilon_{\max} = \max_{\tau=1, \dots, T} \varepsilon_\tau$ . Так как в силу квазирегулярности функции  $p(k, \tau)$

$\varepsilon_\tau \rightarrow 0$  при  $n \rightarrow \infty$ , то  $\varepsilon_{\mathcal{A}_2} \rightarrow 0$  при  $n \rightarrow \infty$ . Теорема 8 доказана.

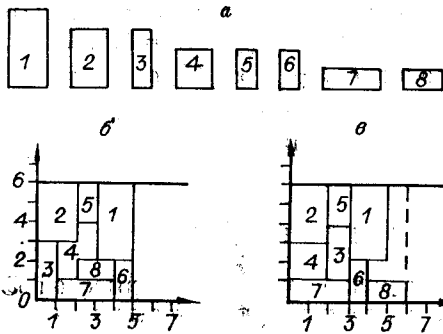


Рис. 2. (а) Список предметов:  $\mathcal{R} = \{(2,4), (2,3), (1,3), (2,2), (1,2), (1,2), (3,1), (2,1)\}$ . (б) Оптимальное решение задачи 1':  $\eta^* = \{3,0,0, 1,2,4, 1,2\}$ . (в) Оптимальное решение задачи упаковки в полосу.

Задачу 1 с одним ограниченным ресурсом нескладированного типа и работами, имеющими постоянные функции потребления ресурсов будем отмечать штрихом.

Заметим, что двумерная задача упаковки в полосу является оценочной сверху для задачи 1' в случае независимых работ. Это следует из примера списка из 8 предметов, представленных прямоугольниками на рис. 2 при упаковке их в полосу с высотой 6. Минимальная ширина полосы в задаче 1' и двумерной задаче упаковки в полосу соответственно равна 5 и 6. Отличие минимальных значений целевых функций обеих задач для указанного списка составляет 20%. Тем не менее из теоремы 8 легко получаем следующее утверждение.

**Теорема 9.** Если  $p(k, \tau)$  квазирегулярна, то последовательное применение алгоритма  $\mathcal{A}_1$  к спискам  $S_\tau$ ,  $\tau = 1, \dots, T$ , приводит к асимптотически точному решению задачи 1' с независимыми работами.

Рассмотрим теперь задачу 1' в случае сетевой модели, задаваемой ориентированным ациклическим графом  $G = (X, U)$ . Рангом  $R_G$  сетевой модели назовем максимальный ранг вершины. Очевидно,  $R_G = \rho_\beta$ , где  $\beta$  — выходное событие сетевой модели. Пусть  $(U_\rho)$  ( $\rho = 1, \dots, R_G$ ) — разбиение множества работ  $U$  на подмножества  $U_\rho$ , в каждое из которых входят работы с концевыми вершинами ранга  $\rho$ . Пусть  $\mathcal{K}_n$  — класс задач 1', для которых справедливо следующее: а)  $|U_\rho| \rightarrow \infty$  при  $n = |U| \rightarrow \infty$ ; б) работам  $u \in U$  приписываются веса  $a_u = k$  и длительности  $\tau_u = \tau$  случайно и независимо друг от друга согласно распределению  $p(k, \tau)$ . Из теоремы 7 вытекает

**Теорема 10.** Для задачи 1' из класса  $\mathcal{K}_n$  в случае квазирегулярности  $p(k, \tau)$  последовательное применение алгоритма  $\mathcal{A}_1$  к подмножествам работ  $U_{\rho,\tau} = \{u | \tau_u = \tau, u \in U_\rho\}$  приводит к асимптотически точному расписанию.

## § 8. О ПРОГРАММНО-МАТЕМАТИЧЕСКОМ ОБЕСПЕЧЕНИИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПЛАНИРОВАНИЯ КРУПНОМАСШТАБНЫХ ПРОЕКТОВ

Описанные выше алгоритмы решения задачи календарного планирования в условиях ограниченных ресурсов явились основой для построения программного комплекса под общим названием «СИБИРЬ». Комплекс включает в себя:

— программу «СИБИРЬ-80», реализованную в системе АЛГОЛ БЭСМ-6;

— дальнейшее развитие этой программы в виде пакета программ «НЕГА», позволяющего организовать более эффективный режим эксплуатации программного комплекса в системе АЛГОЛ БЭСМ-6;

— программный комплекс «СИБИРЬ-ЕС» в системе ОС ЕС ЭВМ;

— информационную базу о задачах планирования хозяйственного освоения зоны БАМ и развития Западно-Сибирского нефтегазового комплекса.

Разработка проводилась в рамках сотрудничества с ИЭиОПП и Главным управлением информации при Совмине СССР. Информационное обеспечение разрабатывается в ИЭиОПП. Этим же институтом осу-

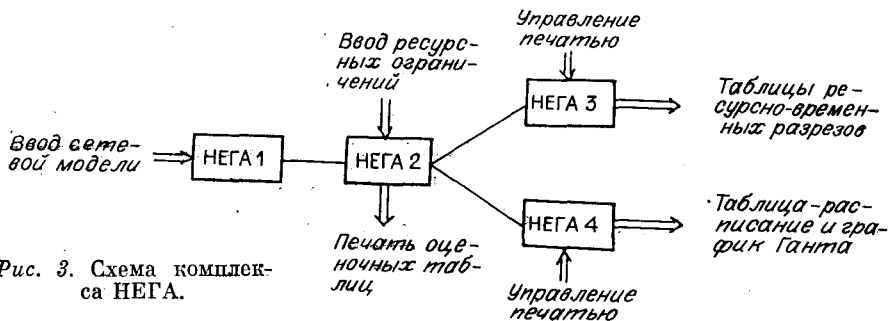


Рис. 3. Схема комплекса НЕГА.

ществляется анализ результатов расчетов по большим сетевым моделям.

Приведем краткое описание программного комплекса «НЕГА», применение которого рассчитано на информационные массивы, учитывающие до 100 отраслей, 100 территориально-промышленных комплексов (промузлов), 50 зон (подрядчиков) и 100 номенклатур ресурсов. Число работ и событий, длительность периода планирования и другие параметры ограничиваются только возможностями используемой ЭВМ и внешних устройств хранения информации.

Комплекс включает в себя следующие четыре основные программы:

○ НЕГА 1 — ввод, контроль правильности задания и преобразование исходной информации к рабочему виду;

○ НЕГА 2 — ввод ресурсных ограничений, построение искомого расписания и вычисление верхних и нижних оценочных таблиц потребления ограниченных ресурсов;

○ НЕГА 3 — печать таблиц общей динамики ресурсов и ресурсно-временных резервов по отраслям, промузлам (ТПК) и зонам (подрядчикам);

○ НЕГА 4 — печать таблицы-расписания и графика Ганта.

Взаимосвязь программ комплекса изображена на рис. 3 с помощью одинарных стрелок. Двойные стрелки указывают на наличие ввода исходных данных и управляющих параметров и вывода результатов счета.

Промежуточные результаты счета программ записываются на внешнюю память (магнитный диск). Предусмотрена возможность одновременной работы с несколькими вариантами информационного задания.

Для учета дополнительных ограничений на ресурсы по каждой зоне (совокупности промузлов) разработана модификация программы НЕГА 2 под названием НЕГА 22.

Более подробные сведения о комплексе «СИБИРЬ» приведены в [3].

Ниже мы остановимся на некоторых важных, на наш взгляд, моментах, возникающих в ходе построения программного комплекса.

Для каждой фактической работы  $u \in U$ , сетевой модели  $G = (X, U)$  задается информация о совокупности профилей  $\{\Pi_u, u \in U\}$  (см. § 2, формула (4)). Однако прямое описание временной зависимости ресурсов каждого профиля  $\pi \in \Pi_u$  с помощью своей отдельной функции — эпюры  $r_\pi(t)$  ( $0 \leq t \leq \tau_\pi^{np}$ ) оказывается неэкономным и затруднительным как с точки зрения задания исходных данных, так и с точки зрения хранения и обработки этой информации.

Компактное и вместе с тем унифицированное описание ресурсно-временных характеристик отдельного профиля достигается заданием пары параметров  $(a_\pi, \gamma_\pi)$  с использованием так называемого базового множества  $\mathcal{R}$  нормализованных функций

$$r_\gamma^0: [0, 1] \rightarrow \mathbb{R}^+, \int_0^1 r_\gamma^0(\xi) d\xi = 1, \quad r_\gamma^0(\xi) \geq 0, \quad \gamma = 1, \dots, \Gamma.$$

При этом параметр  $a_\pi$  равен объему задействованных ресурсов профиля

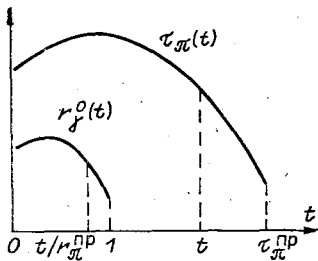


Рис. 4.

чительно меньше общего числа профилей фактических работ в сетевой модели. При этом возникает задача выбора в некотором смысле наилучшего базового множества  $\mathcal{R}$  из множества нормализованных функций, соответствующих совокупности исходных эпюр-профилей  $r_\pi(t)$  ( $0 \leq t \leq \tau_\pi^{\text{np}}$ ,  $\pi \in \Pi_u$ ,  $u \in U_1$ ). Это приводит к формулировке специальной задачи стандартизации (см. § 12).

Для более полного описания ресурсно-временных зависимостей в программном комплексе предусмотрено задание дополнительного параметра категории ресурса, — позволяющего учитывать динамику потребления (производства) ресурсов профилей работ вне интервалов  $[0, \tau_\pi^{\text{np}}]$  их описания.

Следует также упомянуть о многомодульном принципе построения сетевой модели, идея реализации которого исходит из естественного разбиения множества объектов сетевой модели на некоторое число классов, каждый из которых содержит производственные объекты, сходные между собой по технологии выполнения. Указанный принцип предполагает использование конечного набора типовых сетевых графиков (стандартных модулей), каждый из которых моделирует выполнение некоторого гипотетического объекта. Эффективность их применения определяется возможностью адекватного описания процесса планирования любого конкретного производственного объекта в сетевой модели с помощью по крайней мере одного из таких стандартных модулей. В частности, информационная база, разработанная в ИЭиОПП СО АН для задачи развития Западно-Сибирского нефтегазового комплекса, предусматривает задание любого конкретного объекта с помощью одного из шести стандартных модулей. Для задания технологии выполнения конкретного объекта достаточно указания только дуг-работ соответствующего стандартного модуля ненулевой длительности (в случае фактических работ указываются и характеристики профилей). Неуказанные дуги по умолчанию считаются фиктивными, что позволяет еще на предварительном этапе удалить из модели большую часть излишних фиктивных дуг и, возможно, лишних вершин-событий.

## § 9. О ЗАДАЧАХ РАЗМЕЩЕНИЯ

При планировании крупномасштабных проектов в районах пионерного освоения задачи размещения на транспортных сетях занимают исключительно важное место. Транспортные сети в этих районах часто имеют почти древесный характер либо представляют из себя сильно вытянутые конфигурации с локальными подсетями произвольного вида.

Одной из хорошо разработанных математических моделей для решения задач размещения на отрезке (не обязательно прямолинейном) является задача о ближайшем соседе [17—19].

Задачу ближайшего соседа (ЗБС) можно трактовать как задачу разбиения линейного объекта (железнодорожной магистрали, автодороги, линии газо- или нефтепровода и т. п.) на участки «обслуживания». При этом совокупность участков должна представлять из себя разбиение

$\pi$  в интервале  $[0, \tau_\pi^{\text{np}}]$ :  $a_\pi = \int_0^{\tau_\pi^{\text{np}}} r_\pi(t) dt$ , а исходная эпюра-функция этого профиля (рис. 4) восстанавливается по значениям параметров  $\tau_\pi^{\text{np}}$ ,  $a_\pi$  и  $\gamma_\pi \in \{1, \dots, \Gamma\}$ :  $r_\pi(t) = a_\pi \gamma_\pi^0 (t/\tau_\pi^{\text{np}})$ ,  $0 \leq t \leq \tau_\pi^{\text{np}}$ .

Нормализованно-функциональное задание исходной информации о ресурсно-временных характеристиках работ имеет смысл, если число членов  $\Gamma$  в множестве  $\mathcal{R}$  значи-



всей трассы, а затраты по каждому участку зависят от свойств трассы и характера ее прокладки в пределах данного участка и, таким образом, полностью определяются границами этого участка.

Пусть определена неотрицательная функция  $f(x, y)$ ,  $0 \leq x \leq y \leq n$ , где  $x, y$  — целочисленные точки сегмента  $[0, n]$ . Величину  $f(x, y)$  будем понимать как затраты на обслуживание участка  $(x, y]$ . Для любого разбиения сегмента  $[0, n]$  на  $m$  интервалов точками  $x_1, x_2, \dots, x_m$ ,  $0 = x_0 < x_1 < \dots < x_m = n$ , определим целевую функцию

$$W_m(x_1, \dots, x_m) = \sum_{i=1}^m f(x_{i-1}, x_i). \quad (9)$$

ЗБС при фиксированном числе  $m$  интервалов разбиения (ЗБС<sub>*m*</sub>) записывается в виде

$$\sum_{i=1}^m f(x_{i-1}, x_i) \rightarrow \min_{(x_i)}, \quad (10)$$

$$0 = x_0 < x_1 < \dots < x_m = n. \quad (11)$$

К ЗБС<sub>*m*</sub> применима стандартная схема динамического программирования, использующая рекуррентные соотношения

$$S_i(j) = \begin{cases} f(0, j), & j = 0, \dots, n, \text{ при } i = 1, \\ \min \{S_{i-1}(k) + f(k, j) \mid k = 0, \dots, j\}, & j = 0, \dots, n, i = 2, \dots, m, \end{cases} \quad (12)$$

где  $S_i(j)$  — минимальное значение целевой функции в ЗБС<sub>*i*</sub>, рассматриваемой на сегменте  $[0, j]$ . При этом  $T_{\mathcal{A}} \sim mn^2$ ,  $\Pi_{\mathcal{A}} \sim mn$ . В ЗБС с оптимизируемым числом интервалов разбиения (ЗБС\*)

$$m > 0 \quad (13)$$

помимо оптимального разбиения требуется найти и оптимальное количество  $m^*$  интервалов разбиения.

ЗБС\* решается алгоритмом с оценками  $T_{\mathcal{A}} \sim n^2$  и  $\Pi_{\mathcal{A}} \sim n$  с использованием рекуррентных соотношений

$$S(0) = 0, S(j) = \min \{S(k) + f(k, j) \mid k = 0, \dots, j\}, \quad j = 1, \dots, n, \quad (14)$$

где  $S(j)$  — минимальное значение целевой функции в ЗБС\* на отрезке  $[0, j]$ ,  $j = 0, \dots, n$ .

Будем говорить, что функция  $f$  удовлетворяет условию  $\gamma$  [20], если для любых точек  $k_1 \leq j_1 \leq j_2 \leq k_2$  из сегмента  $[0, n]$  (рис. 5) выполнено неравенство

$$f(k_1, k_2) + f(j_1, j_2) \geq f(k_1, j_2) + f(j_1, k_2).$$

При существовании смешанной производной функции  $f(x, y)$  по  $x$  и по  $y$  условие  $\gamma$  эквивалентно неположительности этой производной:  $f_{xy}(x, y) \leq 0$ .

ЗБС с функцией  $f$ , удовлетворяющей условию  $\gamma$ , будем помечать верхним индексом  $\gamma$  (ЗБС<sub>*m*</sub> <sup>$\gamma$</sup>  или ЗБС\* <sup>$\gamma$</sup> ).

**Теорема 11** [19, 20]. *Последовательность  $W_m^*$  минимумов целевой функции в ЗБС<sub>*m*</sub> <sup>$\gamma$</sup>  вытукла по  $m$ :*

$$W_{m-1}^* + W_{m+1}^* \geq 2W_m^*, \quad m = 2, 3, \dots, n-1.$$

Сформулируем утверждения, позволяющие построить алгоритм решения ЗБС<sub>*m*</sub> <sup>$\gamma$</sup>  с оценками  $T_{\mathcal{A}} \sim mn \log_2 n$ ,  $\Pi_{\mathcal{A}} \sim n$ , что дает существенный выигрыш по сравнению со стандартной схемой динамического программирования как по времени (в  $n/\log_2 n$  раз), так и по требуемой памяти (в  $m$  раз) [18].

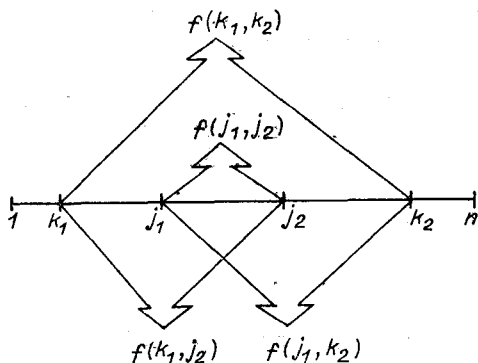


Рис. 5.

Пусть  $x(j)$  и  $x_m(j)$ ,  $m > 1$ , — решения функциональных уравнений (14) и (12) соответственно,  $j = 1, \dots, n$ .

**Теорема 12.** При фиксированном  $x(j_1)$  в ЗБС\* для всякого  $j_2$  ( $j_2 > j_1$ ) найдется такое решение  $x(j_2)$ , что  $x(j_1) \leq x(j_2)$ .

**Следствие 4.** Рекуррентные соотношения для решения ЗБС\* можно записать в виде

$$S(0) = 0; S(j) = \min \{S(k) + f(k, j) | x(j-1) \leq k \leq j\}, j = 1, \dots, n.$$

Упорядоченность решений  $x(1) \leq x(2) \leq \dots \leq x(n)$  при этом ведет к сокращению перебора при вычислении таблиц  $\{S(j), j = 1, \dots, n\}$ .

**Теорема 13.** Пусть при фиксированных  $j_1, j_2$  ( $0 \leq j_1 \leq j_2 \leq n$ ) известны решения  $x_m(j_1) \leq x_m(j_2)$ . Тогда для любого  $j$  ( $j_1 \leq j \leq j_2$ ) в ЗБС\* найдется такое решение  $x_m(j)$ , что  $x_m(j_1) \leq x_m(j) \leq x_m(j_2)$ .

**Теорема 14** (о чередовании оптимальных разбиений). При выполнении условия  $\gamma$  для любого  $m$ -оптимального разбиения  $(x_1, \dots, x_m)$  найдется чередующееся с ним  $(m+1)$ -оптимальное разбиение  $(y_1, \dots, y_{m+1})$ :

$$0 = x_0 \leq y_1 \leq x_1 \leq y_2 \leq \dots \leq x_{m-1} \leq y_m \leq x_m = y_{m+1} = n.$$

Важную роль при построении экономного алгоритма решения ЗБС\* играет следующее утверждение:

**Теорема 15.** Пусть  $(x_1, \dots, x_m)$  —  $m$ -оптимальное разбиение в ЗБС\*. Тогда для любых  $i, 1 \leq i < m-1$ , и  $j, x_i \leq j \leq x_{i+1}$ , найдется такое  $x_{i+1}(j)$ , что  $x_{i-1} \leq x_{i+1}(j) \leq x_i$ .

#### § 10. ЗАДАЧА РАЗМЕЩЕНИЯ С ОБЛАСТЯМИ ОБСЛУЖИВАНИЯ, СВЯЗНЫМИ ОТНОСИТЕЛЬНО АЦИКЛИЧЕСКОЙ СЕТИ

Рассмотрим задачу размещения в следующей математической постановке:

$$\sum_{i \in \mathfrak{M}} g_i^0 x_i + \sum_{i \in \mathfrak{M}} \sum_{j \in \mathfrak{N}} \varphi_j g_{ij} x_{ij} \rightarrow \min_{(x_i)(x_{ij})}, \quad (15)$$

$$\sum_{i \in \mathfrak{M}} x_{ij} = 1 \quad (j \in \mathfrak{N}), \quad (16)$$

$$x_{ij} \leq x_i \quad (i \in \mathfrak{M}, j \in \mathfrak{N}), \quad (17)$$

$$x_i, x_{ij} \in \{0, 1\} \quad (i \in \mathfrak{M}, j \in \mathfrak{N}), \quad (18)$$

где  $\mathfrak{M} = \{1, \dots, m\}$  — множество возможных пунктов производства однородного продукта;  $\mathfrak{N} = \{1, \dots, n\}$  — множество пунктов спроса;  $\varphi_j \geq 0$  — объем спроса в пункте  $j \in \mathfrak{N}$ ;  $g_i^0 \geq 0$  — затраты на размещение предприятия в пункте производства  $i \in \mathfrak{M}$ ;  $g_{ij} \geq 0$  — стоимость доставки единицы продукта из пункта производства  $i \in \mathfrak{M}$  в пункт спроса  $j \in \mathfrak{N}$ .

Переменная выбора  $x_i$  равна 1 при размещении производства в пункте  $i$  и 0 в противном случае; переменная назначения  $x_{ij}$  равна 1, если  $j$ -й пункт спроса удовлетворяется из  $i$ -го пункта размещения, и 0 в противном случае.

В общем случае задача (15) — (18) относится к числу труднорешаемых проблем [5]. В ряде частных случаев построены эффективные методы [21—23]. Ниже мы рассмотрим класс задач, в которых матрице транспортных затрат  $(g_{ij})$  ( $i \in \mathfrak{M}, j \in \mathfrak{N}$ ) может быть сопоставлена ациклическая сеть  $G = (\mathfrak{N}, E)$ , где  $\mathfrak{N} = \{1, \dots, n\}$  — множество вершин,  $E =$

$= \{e_k | k = 1, \dots, n-1\}$  — множество ребер. Вершины сети соответствуют пунктам спроса. В. А. Трубин [21] для случая задачи размещения (15) — (18) на ациклической сети  $G = (\mathfrak{N}, E)$  при  $\mathfrak{M} = \mathfrak{N}$  построил алгоритм с трудоемкостью  $O(n^3)$  в предположениях, что  $g_{ij}$  попарно различны для любых  $i \neq j$  и равны сумме длин ребер сети, входящих в цепь, соединяющую вершины  $i$  и  $j$  ( $i, j \in \mathfrak{N}$ ).

В [22] рассматривается более широкий класс задач размещения на ациклической сети. В случае существования в оптимальном решении областей обслуживания, связанных относительно ациклической сети  $G$ , построен точный алгоритм  $\mathcal{A}$  с трудоемкостью  $O(mn)$ . В частности, задачу из [21] этот алгоритм решает за  $O(n^2)$  действий. Для описания алгоритма использована специальная (сегментная) нумерация вершин сети, отыскиваемая за  $O(n)$  действий. Решение исходной задачи (15) — (18) осуществляется путем сведения ее к некоторой оценочной задаче, для которой справедливы рекуррентные соотношения, используемые для построения алгоритма  $\mathcal{A}$ :

$$f(j) = \min_{i \in \mathfrak{M}} S_i(j) \quad (j \in \mathfrak{N}), \quad (19)$$

где

$$S_i(j) = g_i^0 + \varphi_j g_i + \sum_{k \in \mathfrak{N}_j^0} \min \{f(k), S_i(k) - g_i^0\} \quad (i \in \mathfrak{M}); \quad (20)$$

$\mathfrak{N}_j^0$  — множество сыновей вершины  $j$  в сегментно-занумерованной ациклической сети  $G$ :  $\mathfrak{N}_j^0 = \{k | k \in \mathfrak{N}, (j, k) \in E, k > j\}$ ,  $j \in \mathfrak{N}$ . При этом величина  $f(1)$  есть минимум целевой функции исходной задачи размещения (15) — (18).

**Определение.** Матрица  $(g_{ij})$  ( $i \in \mathfrak{M}$ ,  $j \in \mathfrak{N}$ ) обладает свойством связности относительно сети  $G$ , если для любой пары пунктов размещения  $i, k \in \mathfrak{M}$  существует такое разбиение  $\{\mathfrak{N}^{(i)}, \mathfrak{N}^{(k)}\}$  вершин сети  $G$ , что  $g_{ij} \leq g_{kj}$  ( $j \in \mathfrak{N}^{(i)}$ ),  $g_{kj} \leq g_{ij}$  ( $j \in \mathfrak{N}^{(k)}$ ) и подграфы, порожденные множествами вершин  $\mathfrak{N}^{(i)}$ ,  $\mathfrak{N}^{(k)}$ , связные.

Введенное свойство матрицы является обобщением известного свойства связности [24—29].

**Теорема 16.** Если матрица  $(g_{ij})$  ( $i \in \mathfrak{M}$ ,  $j \in \mathfrak{N}$ ) обладает свойством связности относительно ациклической сети  $G$ , то существует оптимальное решение задачи размещения (15) — (18) с совокупностью связанных областей обслуживания.

Результаты, полученные для задачи (15) — (18), нетрудно распространить на случай нелинейной зависимости функции производственно-транспортных затрат от объема производства.

Пусть  $g_i(v_i)$  — производственная функция  $i$ -го пункта размещения в зависимости от объема производства  $v_i$ . Будем считать, что  $g_i(v_i)$  — функция вогнутая, в нуле равная нулю, кусочно-линейная с числом точек излома не больше  $R_i$ :

$$g_i(v_i) = \begin{cases} 0 & \text{при } v_i = 0, \\ g_{ir}^0 + c_{ir}v_i & \text{при } v_{i,r-1} < v_i \leq v_{ir}, \quad r = 1, \dots, R_i, \end{cases}$$

где  $R_i$  — число диапазонов линейности функции  $g_i(v_i)$ ;  $v_{ir}$  — границы диапазонов линейности,  $v_{i0} = 0$ ,  $v_{iR_i} = \infty$ ;  $g_{ir}^0$  и  $c_{ir}$  — соответственно постоянная составляющая и стоимость производства единицы продукции в  $i$ -м пункте производства в  $r$ -м диапазоне линейности ( $r = 1, \dots, R_i$ ,  $i \in \mathfrak{M}$ ).

Без ограничения общности имеем

$$\begin{aligned} 0 &\leq g_{i1}^0 < g_{i2}^0 < \dots < g_{iR_i}^0, \\ c_{i1} &> c_{i2} > \dots > c_{iR_i}, \\ 0 &= v_{i0} < v_{i1} < \dots < v_{iR_i} = \infty. \end{aligned}$$

Нелинейная задача размещения записывается в виде

$$\sum_{i \in \mathfrak{M}} g_i^0(v_i) + \sum_{j \in \mathfrak{N}} \sum_{i \in \mathfrak{M}} \sum_{r=1}^{R_i} \varphi_j g_{ij} x_{ij}^r \rightarrow \min_{(x_i^r)(x_{ij}^r)} \quad (21)$$

$$\sum_{i \in \mathfrak{M}} \sum_{r=1}^{R_i} x_{ij}^r = 1 \quad (j \in \mathfrak{N}), \quad (22)$$

$$x_{ij}^r \leq x_i^r \quad (r = 1, \dots, R_i, i \in \mathfrak{M}, j \in \mathfrak{N}), \quad (23)$$

$$x_i^r, x_{ij}^r \in \{0, 1\} \quad (r = 1, \dots, R_i, i \in \mathfrak{M}, j \in \mathfrak{N}), \quad (24)$$

где  $x_i^r, x_{ij}^r$  — переменные выбора и назначения, соответствующие  $r$ -му диапазону линейности ( $r = 1, \dots, R_i, i \in \mathfrak{M}, j \in \mathfrak{N}$ ).

Алгоритм решения нелинейной задачи размещения (21) — (24) использует рекуррентные соотношения, аналогичные (19) — (20):

$$f(j) = \min_{i \in \mathfrak{M}} \min_{r=1, \dots, R_i} S_{ir}(j) \quad (j \in \mathfrak{N}), \quad (25)$$

$$S_{ir}(j) = g_{ir}^0 + \varphi_j g_{ij}^r + \sum_{k \in \mathfrak{N}_j^0} \min \{f(k), S_{ir}(k) - g_{ir}^0\} \\ (r = 1, \dots, R_i, i \in \mathfrak{M}, j \in \mathfrak{N}), \quad (26)$$

где  $g_{ij}^r = c_{ir} + g_{ij}$  ( $r = 1, \dots, R_i, i \in \mathfrak{M}, j \in \mathfrak{N}$ ).

При тех же требованиях к матрице ( $g_{ij}$ ) ( $i \in \mathfrak{M}, j \in \mathfrak{N}$ ), что и в линейной задаче, алгоритм, использующий рекуррентные соотношения (25) — (26), решает нелинейную задачу за время  $O(mnR)$ , где  $R = \max_{i \in \mathfrak{M}} R_i$ .

Программная реализация алгоритма выполнена в системе АЛГОЛ БЭСМ-6. Вычислительные эксперименты осуществлены на задачах размещения баз материально-технического снабжения в зоне БАМ [4].

#### § 11. ЗАДАЧА РАЗМЕЩЕНИЯ НА СЕТИ С ЦЕНТРАЛЬНО-СВЯЗНЫМИ ОБЛАСТЯМИ ОБСЛУЖИВАНИЯ

Рассмотрим задачу размещения вида (15) — (18) на сети  $G = (\mathfrak{N}, E)$ , не являющейся ациклической, с множеством возможных пунктов размещения  $\mathfrak{M}$ , принадлежащим множеству пунктов спроса  $\mathfrak{N}$ :  $\mathfrak{M} \subset \mathfrak{N}$  [23]. Предполагаем заданными неотрицательные длины ребер сети, с помощью которых могут быть определены кратчайшие цепи между вершинами сети.

Задачу (15) — (18) можно записать в эквивалентном виде

$$\sum_{i \in I_\sigma} g_i^0 + \sum_{j \in \mathfrak{N}} \varphi_j g_{ij} \rightarrow \min_{\sigma} \quad (27)$$

где решением является вектор назначений  $\sigma = (i_j)$  ( $j \in \mathfrak{N}$ );  $i_j$  — номер предприятия, обслуживающего  $j$ -й пункт спроса ( $i_j \in \mathfrak{M}, j \in \mathfrak{N}$ );  $I_\sigma = \cup_{j \in \mathfrak{N}} \{i_j\}$  — набор предприятий, соответствующих вектору назначений  $\sigma$ ,  $I_\sigma \subset \mathfrak{M}$ .

Область обслуживания  $\mathfrak{N}_i(\sigma) = \{j | i_j = i, j \in \mathfrak{N}\}$  назовем *связной относительно сети  $G$* , если подграф, порожденный множеством вершин этой области, связан, и *центральной*, если обслуживающее предприятие содержится внутри этой области. Будем говорить, что матрица ( $g_{ij}$ ) ( $i \in \mathfrak{M}, j \in \mathfrak{N}$ ) удовлетворяет *цепочечному условию на сети  $G$*  (является *Ц-матрицей*), если из  $g_{ij} < g_{kl}$  при  $l \in \mathfrak{N}, i, k \in \mathfrak{M}$  следует  $g_{ij} < g_{kj}$  для любой вершины  $j \in \mathfrak{N}$ , принадлежащей кратчайшей цепи из  $i$  в  $l$ . Таковой, в частности, является матрица ( $\tilde{g}_{ij}$ ) ( $i \in \mathfrak{M}, j \in \mathfrak{N}$ ) с компонентами  $\tilde{g}_{ij} =$

$= c_i + \rho_{ij}$ , где  $c_i \geq 0$ ,  $\rho_{ij} \geq 0$  — сумма длин ребер в кратчайшей цепи из вершины  $i$  в вершину  $j$ .

**Теорема 17.** Для задачи размещения (27) на сети  $G$  с Ц-матрицей  $(g_{ij})$  ( $i \in \mathfrak{M}$ ,  $j \in \mathfrak{N}$ ) существует оптимальное решение  $\sigma$  с совокупностью центрально-связанных областей применения.

Назовем основой сети  $G$  максимальный связный подграф  $D \subset G$ , ни одно ребро которого не принадлежит никакому циклу сети  $G$ ;  $\mathfrak{N}(D)$  — множество вершин этого подграфа. Выберем произвольную вершину  $v \in \mathfrak{N}(D)$  в качестве корневой вершины сети  $G$ . Обозначим через  $\mathfrak{N}_j$  множество вершин сети  $G$ , лежащих на продолжении цепи из корневой вершины в вершину  $j$  ( $v, j \in \mathfrak{N}(D)$ ). Максимальный циклический подграф сети  $G$ , точки сочленения которого не принадлежат основе  $D$  этой сети, назовем  $D$ -квазиблоком.

Заметим, что  $D$ -квазиблок определяется независимо от того, какая из вершин основы взята в качестве корневой. Далее, считая основу  $D$  сети  $G$  зафиксированной, будем пользоваться более коротким наименованием «квазиблок».

Обозначим  $\{(B_j^k) (k \in A_j)\} (j \in \mathfrak{N}(D))$  — совокупность квазиблоков сети  $G$ ;

$B_j^k$  — некоторый квазиблок, инцидентный вершине  $j \in \mathfrak{N}(D)$ ;

$(B_j^k) (k \in A_j)$  — совокупность квазиблоков, инцидентных одной и той же вершине  $j \in \mathfrak{N}(D)$ ;

$\mathfrak{N}_j^k$  — множество вершин квазиблока  $B_j^k (k \in A_j, j \in \mathfrak{N}(D))$ .

Псевдодревесным назовем такой квазиблок  $B_j^k$ , у которого подграф, порожденный множеством вершин  $\mathfrak{N}_j^k \setminus \{j\}$ , является ациклическим. Квазиблоком с логарифмически ограниченным числом предприятий обслуживания назовем квазиблок  $B_j^k$ , в котором число возможных мест размещения не превышает  $\log_2 n$ . Сеть  $G$  назовем сегментно-занумерованной относительно ее основы  $D$  с выбранной корневой вершиной  $v$ , если для любой вершины  $j$  основы  $D$  каждое из множеств  $\mathfrak{N}_j$  и  $\mathfrak{N}_j^k \setminus \{j\}$  ( $k \in A_j$ ) есть целочисленный сегмент, причем  $j$  — минимальная вершина в  $\mathfrak{N}_j$  ( $j \in \mathfrak{N}(D)$ ).

На рис. 6 изображен пример сегментно-занумерованной сети с корневой вершиной  $v = 1$ ,  $\mathfrak{N}(D) = \{1, 2, 3, 9, 10, 18, 19, 24, 25, 26, 27, 28\}$  и четырьмя квазиблоками с множествами вершин  $\mathfrak{N}_3^1 = \{3, \dots, 8\}$ ,  $\mathfrak{N}_{10}^1 = \{10, 11, 12\}$ ,  $\mathfrak{N}_{10}^2 = \{10, \dots, 17\} \setminus \{11, 12\}$ ,  $\mathfrak{N}_{19}^1 = \{19, \dots, 23\}$ ; 2-й и 4-й квазиблоки являются псевдодревесными.

Сегментная нумерация вершин сети осуществляется с трудоемкостью, линейной от числа ребер сети.

**Теорема 18.** В случае сети  $G$ , содержащей только псевдодревесные квазиблоки, задача размещения с Ц-матрицей  $(g_{ij})$  ( $i \in \mathfrak{M}$ ,  $j \in \mathfrak{N}$ ) может быть решена с трудоемкостью, ограниченной величиной  $O(m^2n + N)$ , где  $N$  — число ребер сети.

**Теорема 19.** В случае сети  $G$ , состоящей только из логарифмически ограниченных квазиблоков, точное решение задачи размещения с Ц-матрицей  $(g_{ij})$  ( $i \in \mathfrak{M}$ ,  $j \in \mathfrak{N}$ ) может быть получено за  $O(mn^2)$  действий.

**Следствие 5.** Если сеть  $G$  содержит квазиблоки только псевдодревесного и (или) логарифмически-ограниченного типа, то задача размещения с Ц-матрицей  $(g_{ij})$  ( $i \in \mathfrak{M}$ ,  $j \in \mathfrak{N}$ ) может быть решена за  $O(mn(t+n))$  действий.

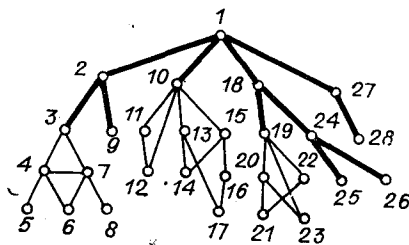


Рис. 6.

**§ 12. О ЗАДАЧАХ СТАНДАРТИЗАЦИИ  
(ВЫБОРА ТИПАЖА И СОСТАВА ОБОРУДОВАНИЯ)**

При планировании мероприятий по хозяйственному освоению новых территорий возникает проблема создания соответствующего парка машин, механизмов и т. п. Наиболее подробное изложение основных математических моделей и методов решения задач выбора типажа оборудования, далее для краткости — задач стандартизации (ЗС), содержится в работе [24]. Сформулируем так называемую базовую модель ЗС:

$$L(\bar{\mathfrak{M}}) = \sum_{i \in \bar{\mathfrak{M}}} g_i^0 + \sum_{j \in \mathfrak{R}} \min_{i \in \bar{\mathfrak{M}}} g_{ij} \rightarrow \min, \quad (28)$$

$$|\bar{\mathfrak{M}}| \leq N_0, \quad (29)$$

где  $\mathfrak{M} = \{1, \dots, m\}$  — множество, соответствующее перечню типов изделий;  $\mathfrak{R} = \{1, \dots, n\}$  — множество, соответствующее совокупности видов спроса;  $g_i^0$  — начальные затраты за ввод в действие изделия  $i$ -го типа ( $i \in \mathfrak{M}$ );  $g_{ij}$  — затраты, связанные с обслуживанием спроса  $j$ -го вида с помощью изделия  $i$ -го типа ( $i \in \mathfrak{M}, j \in \mathfrak{R}$ ).

Сформулированная задача является NP-трудной. Более того, NP-трудной остается задача построения алгоритма  $\mathcal{A}$ , отыскивающего решение  $\bar{\mathfrak{M}}_{\mathcal{A}}$ , удовлетворяющее ограниченному отношению  $L(\bar{\mathfrak{M}}_{\mathcal{A}})/L^* \leq r_{\mathcal{A}}$  для каждой индивидуальной задачи [5]. Тем не менее получен ряд результатов, позволяющих построить эффективные точные методы решения ЗС — при дополнительных предположениях на исходные данные. К таким предположениям в первую очередь следует отнести наиболее хорошо изученные свойства матрицы  $(g_{ij})$  ( $i \in \mathfrak{M}, j \in \mathfrak{R}$ ) — связности и квазивыпуклости [24–30].

Будем писать сокращенно ЗС<sup>+</sup> для задачи стандартизации с неотрицательными исходными данными. Очевидны следующие утверждения.

**Лемма 7.** ЗС с  $g_{ij} \in \mathbf{R}^{m \times n}$  сводится к ЗС с  $g_{ij} \geq 0, i \in \mathfrak{M}, j \in \mathfrak{R}$ , с помощью преобразования  $g_{ij}^{\text{нов}} = g_{ij}^{\text{стар}} - g_j, \text{ где } g_j = \min_{i \in \mathfrak{M}} g_{ij}^{\text{стар}}, j \in \mathfrak{R}$ .

**Лемма 8.** ЗС вида (28) с парой  $(g_i^0)(g_{ij})$  ( $i \in \mathfrak{M}, j \in \mathfrak{R}$ ) сводится к решению ЗС<sup>+</sup> вида (28) с парой  $(g_i^0)(g_{ij})$  ( $i \in \mathfrak{M}^+, j \in \mathfrak{R}$ ), где  $\mathfrak{M}^+ = \{i | g_i^0 > 0, i \in \mathfrak{M}\}, g_{ij} = \min \{g_{ij}; \min_{k \in \mathfrak{M} \setminus \mathfrak{M}^+} g_{kj}\}$  ( $i \in \mathfrak{M}^+, j \in \mathfrak{R}$ ).

**Следствие 6.** ЗС вида (28) сводится к решению ЗС<sup>+</sup> вида (28).

Для ЗС с дополнительным ограничением (29) такое сведение осуществить не удается.

**§ 13. ЗАДАЧИ СТАНДАРТИЗАЦИИ СО СВЯЗНЫМИ МАТРИЦАМИ**

Матрицу  $(g_{ij})$  ( $i \in \mathfrak{M}, j \in \mathfrak{R}$ ) назовем *связной*, если для всякой пары строк  $i, k \in \mathfrak{M}$  разность  $(g_{ij} - g_{kj})$  меняет знак не более одного раза. Связной задачей стандартизации (СЗС) будем называть ЗС со связной матрицей. Аналогично ЗС<sup>+</sup> употребляем сокращение СЗС<sup>+</sup> для СЗС с неотрицательными исходными данными. СЗС<sup>+</sup> вида (28) решается за  $O(mn)$  действий с помощью рекуррентных соотношений (19)–(20), где множество  $\mathfrak{R}_j^0, 1 < j \leq n$ , состоит из единственной вершины  $(j-1)$ . Модификация этих соотношений  $f_N(j) = \min_{1 < i < m} S_{N,i}(j)$  ( $j = 1, \dots, n; N = 1, \dots, N_0$ ),

$$S_{N,i}(j) = \begin{cases} g_i^0 + \sum_{l=1}^j g_{il} & \text{при } N = 1, \\ \min \{g_i^0 + f_{N-1}(j-1); S_{N,i}(j-1)\} + g_{ij} & \text{при } 1 < N \leq N_0. \end{cases}$$

( $i = 1, \dots, m, j = 1, \dots, n$ ) позволяет решить СЗС<sup>+</sup> вида (28) — (29) за  $O(N_0 mn)$  действий.

**Лемма 9.** Если матрица  $(g_{ij})$  ( $i \in \mathfrak{M}, j \in \mathfrak{N}$ ) связана, то и матрицы  $(g'_{ij})$  ( $i \in \mathfrak{M}^+, j \in \mathfrak{N}$ ), определенная в лемме 8, связана.

Тем самым СЗС вида (28) также решается за  $O(mn)$  действий.

Для СЗС вида (28) — (29) в [29] обоснован эффективный точный алгоритм, требующий предварительного лексикографического упорядочения строк матрицы  $(g_{ij})$  ( $i \in \mathfrak{M}, j \in \mathfrak{N}$ ) и использующий рекуррентные соотношения

$$f_{N,i}(j) = \min \{f_{N,i-1}(j); S_{N,i}(j)\}, \quad (30)$$

$$S_{N,i}(j) = \min \{g_{ij}^0 + f_{N-1,i-1}(j); g_{ij} + S_{N,i}(j-1)\} \quad (31)$$

$$(N = 1, \dots, N_0, i = 1, \dots, m, j = 1, \dots, n).$$

В результате своей работы алгоритм находит оптимальное значение целевой функции  $L(\mathfrak{M}^*) = f_{N,m}(n)$  и согласованно-связанное решение  $\{i_N, j_N\}$  ( $N = 1, \dots, N^*$ ), где  $N^* = |\mathfrak{M}^*| \leq N_0$  — число членов в оптимальном наборе  $\mathfrak{M}^* = \{i_1, \dots, i_{N^*}\}$ ,  $i_1 < \dots < i_{N^*}$ , а  $j_1, \dots, j_{N^*}$  — границы связанных областей обслуживания,  $0 = j_0 \leq j_1 \leq \dots \leq j_{N^*} = n$ , так что издание  $i_N$  обслуживает область  $(j_{N-1}, j_N]$  ( $N = 1, \dots, N^*$ ). Время работы алгоритма  $\sim n(N_0 m + r)$ , где  $r$  — оценка трудоемкости упорядочения одного столбца матрицы, равная  $\sim m \log_2 m$  (для произвольных элементов матрицы) и  $r_0$  (при  $g_{ij}$  целых,  $0 \leq g_{ij} < r_0$ ).

#### § 14. ЗАДАЧИ СТАНДАРТИЗАЦИИ С КВАЗИВЫПУКЛЫМИ И ОБОБЩЕННО-КВАЗИВЫПУКЛЫМИ МАТРИЦАМИ

Столбец  $j \in \mathfrak{N}$  матрицы  $(g_{ij})$  ( $i \in \mathfrak{M}, j \in \mathfrak{N}$ ) называют квазिवыпуклым (квазивоогнутым), если для любых  $i_1, i_2, i_3 \in \mathfrak{M}$ ,  $i_1 \leq i_2 \leq i_3$ , справедливо  $g_{i_2,j} \leq \max \{g_{i_1,j}; g_{i_3,j}\}$  ( $g_{i_2,j} \geq \min \{g_{i_1,j}; g_{i_3,j}\}$ ). Матрицу и задачу стандартизации будем называть квазिवыпуклой (квазивоогнутой), если соответствующее свойство имеет место для каждого столбца матрицы.

В. Л. Береснев [28], используя представление квазिवыпуклой ЗС<sup>+</sup> в виде задачи минимизации так называемого правильного полинома от булевых переменных, осуществив сведение последней к СЗС<sup>+</sup>, что позволило построить алгоритмы с оценками трудоемкости  $\sim m(n+m)$  и  $\sim m(n+N_0 m)$  соответственно для постановок (28) и (28) — (29). Однако более перспективным [29] для квазिवыпуклых ЗС оказывается сводимость их к задаче ближайшего соседа (10) — (11) с функцией

$$f(i, k) = \sum_{i < p < q < k} \varphi_{pq} \quad (0 \leq i < k \leq m + 1),$$

где

$$\varphi_{pq} = \sum_{(l,j) \in \Pi_{pq}} (c_l^j - c_{l-1}^j) \quad (1 \leq p \leq q \leq m),$$

$c_1^j \leq \dots \leq c_m^j$  — упорядоченные по неубыванию элементы  $j$ -го столбца матрицы  $(g_{ij})$ ,  $j = 1, \dots, n$ ;

$$\Pi_{pq} = \{(l, j) \mid \mathcal{D}_l^j = [p, q], l = 1, \dots, m, j = 1, \dots, n\}, \quad 1 \leq p \leq q \leq m;$$

$\mathcal{D}_l^j$  — дополнение до множества  $\mathfrak{M}$  лебегова множества

$\{g_{ij} \geq c_l^j\}$  целочисленных точек  $i = 1, \dots, m$ , в которых  $g_{ij}$  не меньше  $c_l^j$ ,  $l = 1, \dots, m, j = 1, \dots, n$ .

В этом случае функция  $f$  удовлетворяет условию  $\gamma$  (см. § 9), и оценка трудоемкости решения задачи (28) имеет тот же порядок, что и при сведении к СЗС, а для задачи (28) — (29) снижается до величины  $O(mn + m^2 + N_0 m \log_2 m)$ .

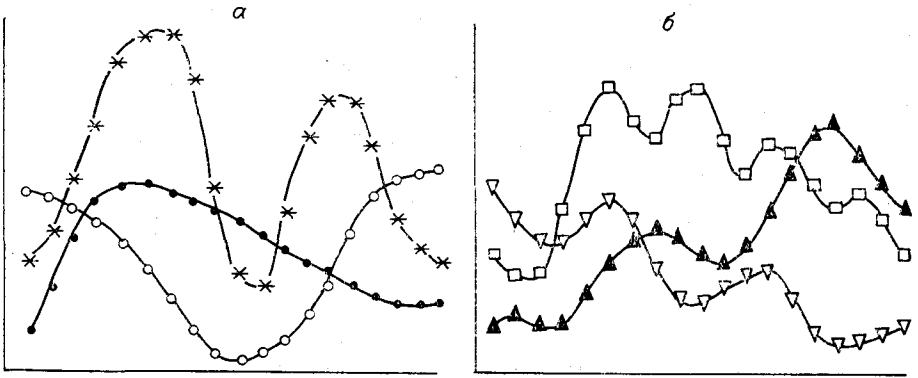


Рис. 7. Примеры функций: (а)  $\circ$  — квазивыпуклая,  $\bullet$  — квазивогнутая,  $*$  — с одним внутренним минимумом; (б) квазивыпуклая справа ( $\nabla$ ) и слева ( $\blacktriangle$ ),  $\square$  — обобщенно-квазивыпуклая.

Опишем более широкий класс ЗС, полиномиально разрешимых за счет сводимости их к квазивыпуклым задачам.

Компоненту связности назовем *внутренней* (относительно множества  $\mathfrak{M}$ ), если она не содержит граничные точки множества  $\mathfrak{M}$ . Матрицу  $(g_{ij})$  ( $i \in \mathfrak{M}, j \in \mathfrak{N}$ ) и соответствующую ей ЗС назовем *квазивыпуклой справа (слева)*, если лебегово множество  $\{g_{ij} \geq c_l^j\}$  для всяких  $l = 1, \dots, \dots, t$  и  $j = 1, \dots, n$  состоит не более чем из двух компонент связности, причем максимальная (минимальная) точка внутренней компоненты связности совпадает с  $\max \{g_{ij} \geq c_l^j\}$  ( $\min \{g_{ij} \geq c_l^j\}$ ).

**Лемма 10.** Если матрица  $(g_{ij})$  ( $i \in \mathfrak{M}, j \in \mathfrak{N}$ ) квазивыпукла справа (слева), то для любого  $k, 1 < k \leq t$ , матрица  $(g_{ij}^k)$  ( $i \in \mathfrak{M}, j \in \mathfrak{N}$ ), где  $g_{ij}^k = \min \{g_{ij}; g_{kj}\}$ , квазивыпукла на множестве  $\mathfrak{M}^{(k)}$ , представляющем из себя сегмент  $[1, k]$  ( $[k, t]$ ).

**Следствие 7.** Квазивыпуклая справа (слева) ЗС путем сведения к последовательности квазивыпуклых ЗС решается за число действий  $O(m^2n + t^3)$  и  $O(m^2n + t^3 + N_0 m^2 \log_2 t)$  (соответственно для постановок (28) и (28) — (29)).

Частным случаем квазивыпуклых справа (или слева) ЗС является квазивыпукло-вогнутая ЗС [30], в которой матрица имеет как квазивыпуклые, так и квазивогнутые столбцы.

Рассмотрим класс ЗС, обобщающих рассмотренные выше квазивыпуклые, квазивогнутые, квазивыпуклые справа (слева) задачи.

Матрицу  $(g_{ij})$  ( $i \in \mathfrak{M}, j \in \mathfrak{N}$ ) и соответствующую ей ЗС назовем *обобщенно-квазивыпуклой*, если для всяких  $l = 1, \dots, t$  и  $j = 1, \dots, n$  лебегово множество  $\{g_{ij} \geq c_l^j\}$  состоит не более, чем из двух компонент связности.

**Лемма 11.** Если матрица  $(g_{ij})$  ( $i \in \mathfrak{M}, j \in \mathfrak{N}$ ) обобщенно-квазивыпуклая, то для всяких  $k, l, 1 \leq k < l \leq t$ , преобразованная матрица  $(g_{ij}^{(k,l)})$  ( $i \in \mathfrak{M}^{(k,l)}, j \in \mathfrak{N}$ ), где  $g_{ij}^{(k,l)} = \min \{g_{ij}; g_{kj}; g_{lj}\}$ , квазивыпукла на множестве  $\mathfrak{M}^{(k,l)} = [k, l]$ .

**Следствие 8.** Обобщенно-квазивыпуклая ЗС путем сведения ее к последовательности квазивыпуклых ЗС решается за время  $\sim t^3(n+t)$  и  $\sim t^3(n+t + N_0 \log_2 t)$  (соответственно для задач (28) и (28) — (29)).

На рис. 7 изображены графики квазивыпуклой, квазивогнутой, квазивыпуклой справа, квазивыпуклой слева и обобщенно-квазивыпуклой функций, а также график функции с одним внутренним минимумом.



§ 15. О ПРИБЛИЖЕННЫХ АЛГОРИТМАХ С ОЦЕНКАМИ  
ДЛЯ РЕШЕНИЯ ЗАДАЧИ СТАНДАРТИЗАЦИИ

В общем случае хорошо себя зарекомендовал метод ветвей и границ [24] с построением так называемого тупикового решения задачи

$$\left\{ \sum_{j \in \mathcal{R}} \min_{i \in \mathcal{M}} (g_{ij} + w_{ij}) \mid \sum_{j \in \mathcal{R}} w_{ij} \leq g_i^0 (i \in \mathcal{M}) \right\} \rightarrow \max_{w_{ij} \geq 0} \quad (32)$$

двойственной к расширению ЗС, полученному из линейно-целочисленной постановки снятием ограничения на целочисленность переменных. Этот метод привлекателен возможностью осуществления приближенной схемы его работы. Кроме того, идея построения тупикового решения была использована при реализации эффективного алгоритма, имеющего трудоемкость  $O(m^2n)$  действий, с апостериорным оцениванием погрешности получаемого значения целевой функции [24].

Особый интерес, однако, представляет построение малотрудоемких статистически эффективных алгоритмов с априорными оценками  $\varepsilon_{\mathcal{A}}$ ,  $\delta_{\mathcal{A}}$  качества решения задачи (28) — (31) [31].

Определим класс ЗС  $(g_{\min}^0(n), g_{\max}^0(n), a_n, b_n)$  задач стандартизации с элементами вектора  $(g_i^0) (i \in \mathcal{M})$  и матрицы  $(g_{ij}) (i \in \mathcal{M}, j \in \mathcal{R})$ , принимающими значения соответственно из числовых отрезков  $[g_{\min}^0(n), g_{\max}^0(n)]$  и  $[a_n, b_n]$ ;  $g_{\min}^0(n)$  и  $a_n$  неотрицательны. При этом предполагаем, что элементы  $g_{ij}$  выбираются из отрезка  $[a_n, b_n]$  случайно и независимо друг от друга с одинаковой функцией распределения.

Опишем малотрудоемкий, вообще говоря, приближенный алгоритм  $\tilde{\mathcal{A}}$  для решения задач из класса ЗС  $(g_{\min}^0(n), g_{\max}^0(n), a_n, b_n)$ .

Этап 1. Если  $m \leq \log_2 n$ , то перебрав не более  $n$  различных подмножеств  $\tilde{\mathcal{M}} \subset \mathcal{M}$ , найдом точное решение и заканчиваем работу алгоритма. В противном случае переходим ко второму этапу алгоритма.

Этап 2. (Случай  $m > \log_2 n$ .)

2.1. Вычисляем параметр  $\tilde{m}$  по формуле

$$\tilde{m} = \begin{cases} \lfloor m_0 \rfloor & \text{при } m \geq m_0, \\ m & \text{при } m < m_0, \end{cases}$$

где  $m_0 = \sqrt{n(b_n - a_n)/g_{\max}^0(n)} - 1$ .

2.2. Отыскиваем подмножество  $\tilde{\mathcal{M}} \subset \mathcal{M}$  из  $\tilde{m}$  первых в порядке неубывания элементов вектора  $(g_i^0) (i \in \mathcal{M})$ :

$$\max_{i \in \tilde{\mathcal{M}}} g_i^0 \leq \min_{i \in \mathcal{M} \setminus \tilde{\mathcal{M}}} g_i^0.$$

2.3. В качестве выходного результата принимаем подмножество  $\tilde{\mathcal{M}}$ , значение целевой функции  $\mathbf{L} = \mathbf{L}(\tilde{\mathcal{M}})$  и вектор назначения  $(i_j) (j \in \mathcal{R})$ , где  $i_j = \arg \min \{g_{ij} \mid i \in \tilde{\mathcal{M}}\} (j \in \mathcal{R})$ . На этом алгоритм  $\tilde{\mathcal{A}}$  заканчивает свою работу.

Нетрудно убедиться в том, что трудоемкость алгоритма ограничена величинами  $O(n^2)$  при  $m \leq \log_2 n$  и  $O(\tilde{m}n + \tilde{m} \log_2 m)$  при  $m > \log_2 n$ . При весьма правдоподобном предположении  $m \leq 2^n$  оценка трудоемкости алгоритма  $\tilde{\mathcal{A}}$  в целом имеет порядок величины  $O(n^2 + n\tilde{m})$ .

**Теорема 20.** Алгоритм  $\tilde{\mathcal{A}}$  гарантирует получение решения  $\mathcal{M}$  задачи из класса ЗС  $(g_{\min}^0(n), g_{\max}^0(n), a_n, b_n)$  со значением целевой функции  $\hat{\mathbf{L}}$ , которое с вероятностью, стремящейся к 1 при  $n \rightarrow \infty$ , не превосходит величины

$$\hat{\mathbf{L}} = \frac{mg_{\max}^0(n)}{b_n - a_n} + (1 + \varepsilon_n^1) \frac{n}{m + 1}, \quad (33)$$

где  $\varepsilon_n^1 = \sqrt{(\ln n)/n} \rightarrow 0$  при  $n \rightarrow \infty$ .

**З а м е ч а н и е.** Число элементов множества  $\tilde{\mathcal{M}}$  в алгоритме  $\tilde{\mathcal{A}}$  выбрано из экстремальных соображений: при таком выборе параметра  $\tilde{m}$ , как нетрудно убедиться из формулы (33), верхняя оценка  $\tilde{L}$  для целевой функции, получаемой в результате работы алгоритма  $\tilde{\mathcal{A}}$ , при малых  $\varepsilon_n^1$  близка к своему минимальному значению.

**Теорема 21.** Алгоритм  $\tilde{\mathcal{A}}$  решения ЗС  $(g_{\min}^0(n), g_{\max}^0(n), a_n, b_n)$ ,  $a_n > 0$ , асимптотически точен, если

$$\frac{b_n - a_n}{a_n} = o(\log_2 n), \quad (34)$$

$$\frac{g_{\max}^0(n)}{a_n} = o\left(\frac{n}{\log_2 n}\right). \quad (35)$$

Далее рассмотрим подкласс ЗС' задач стандартизации с равномерным распределением случайных величин  $g_{ij}$  в интервале  $[a_n, b_n]$  и с параметрами  $m, g_{\min}^0(n), a_n, b_n$ , такими, что

$$m = o(n^\lambda), \quad (36)$$

$$\frac{g_{\min}^0(n)}{b_n - a_n} \geq \psi_n \frac{\ln^2 n}{n}, \quad (37)$$

где  $\lambda$  — константа,  $\lambda \geq 1$ ;  $\psi_n$  — растущая медленнее, чем  $n$ , функция:  $\psi_n \rightarrow \infty$  при  $n \rightarrow \infty$ ,  $\psi_n = o(n)$ .

**Теорема 22.** Применение алгоритма  $\tilde{\mathcal{A}}$  к задаче из класса ЗС'  $(g_{\min}^0(n), g_{\max}^0(n), a_n, b_n)$  дает отношение  $\tilde{L}/L^*$ , которое не превосходит величины

$$r_{\tilde{\mathcal{A}}} \sim \frac{3}{2} \sqrt{g_{\max}^0(n)/g_{\min}^0(n)} \quad (38)$$

с вероятностью, стремящейся к 1 при  $n \rightarrow \infty$ .

**З а м е ч а н и е.** При величине  $m + 1$ , не входящей в интервал  $[\beta m_0, m_0]$ ,  $\beta = (\sqrt{2} - 1)^{1/2} \approx 0,63$ , константа  $3/2$  в формуле (38) может быть заменена на константу  $\sqrt{2}$ .

**Следствие 9.** Для алгоритма  $\tilde{\mathcal{A}}$ , применяемого к задаче из класса ЗС' с одинаковыми элементами вектора  $(g_i^0) (i \in \mathcal{M})$ :  $g_i^0 = g^0, i = 1, \dots, m$ , справедливы оценки  $\varepsilon_{\tilde{\mathcal{A}}}^1, \delta_{\tilde{\mathcal{A}}}^1$ , асимптотически равные соответственно  $1/2$  и  $0$ :  $\varepsilon_{\tilde{\mathcal{A}}}^1 \rightarrow 1/2, \delta_{\tilde{\mathcal{A}}}^1 \rightarrow 0$  при  $n \rightarrow \infty$ .

Целый ряд более сложных постановок задачи выбора типажа оборудования [24], возникающих при планировании крупномасштабных проектов, решаются либо путем сведения к базовой модели ЗС, либо с ее использованием при построении оценочных задач.

## ЛИТЕРАТУРА

1. Гимади Э. Х., Пузынина Н. М., Севостьянов С. В. О некоторых экстремальных задачах реализации крупных проектов типа БАМ // Экономика и мат. методы.— 1979.— Вып. 5.— С. 1017—1020.
2. Gimadi E. Kh. On some optimisation problems of large projects planning and control // IFIP Abstracts. 9th IFIP Conference on Optimization Techniques.— Warsaw, 1979.— P. 78—79.
3. Гимади Э. Х., Пузынина Н. М. Задача календарного планирования крупномасштабного проекта в условиях ограниченных ресурсов: опыт построения математического обеспечения // Управляемые системы.— Новосибирск, 1983.— Вып. 23.— С. 24—32.
4. Гимади Э. Х., Кисельников А. А. и др. Оптимизация развития и размещения баз материально-технического снабжения в зоне БАМ // Материалы IV Всесоюзной конференции по проблемам БАМ, Благовещенск, 1985.— С. 141—154.
5. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи.— М.: Мир, 1982.— 416 с.

6. Гимади Э. Х., Глебов Н. И., Перепелица В. А. Алгоритмы с оценками для задач дискретной оптимизации // Проблемы кибернетики.— М.: Наука, 1975.— Вып. 31.— С. 35—42.
7. Гимади Э. Х., Глебов Н. И., Перепелица В. А. Исследования по теории расписаний // Управляемые системы.— Новосибирск, 1974.— Вып. 12.— С. 3—10.
8. Гимади Э. Х., Перепелица В. А. Статистически эффективный алгоритм выделения гамильтонова контура (цикла) // Дискретный анализ.— Новосибирск, 1973.— Вып. 22.— С. 15—28.
9. Перепелица В. А., Гимади Э. Х. К задаче нахождения минимального гамильтонова контура на графе со взвешенными дугами // Дискретный анализ.— Новосибирск, 1969.— Вып. 15.— С. 57—65.
10. Гимади Э. Х., Перепелица В. А. Асимптотически точный подход к решению задачи коммивояжера // Управляемые системы.— Новосибирск, 1974.— Вып. 12.— С. 35—45.
11. Гимади Э. Х., Залюбовский В. В. Асимптотически точный подход к решению одномерной задачи упаковки в контейнеры // Управляемые системы.— Новосибирск, 1984.— Вып. 25.— С. 48—57.
12. Литтл Дж., Мурти К., Суини Д., Кэрел К. Алгоритм для решения задачи о коммивояжере // Экономика и мат. методы.— 1965.— Вып. 1.— С. 94—107.
13. Боровков А. А. К вероятностной постановке двух экономических задач // Докл. АН СССР.— 1962.— Т. 146, № 5.— С. 983—986.
14. Михалевиц В. С., Кукса А. И. Методы последовательной оптимизации.— М.: Наука, 1983.— 208 с.
15. Johnson D. S., Demers A., Ullman J. D., Garey M. R. and Graham R. L. Worst-case performance bounds for simple one-dimensional packing algorithms // SIAM J. comput.— 1974.— N 3.— P. 299—325.
16. Бернштейн С.Н. Теория вероятностей.— М.: ОНТИ, 1934.— 412 с.
17. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования.— М.: Наука, 1965.— 458 с.
18. Гимадуддинов Э. Х. Об одном классе задач нелинейного программирования // Управляемые системы.— Новосибирск, 1969.— Вып. 3.— С. 101—113.
19. Гимади Э. Х., Глебов Н. И. Экстремальные задачи принятия решений.— Новосибирск: Новосибирский ун-т, 1982.— 80 с.
20. Глебов Н. И. О выпуклых последовательностях // Дискретный анализ.— Новосибирск, 1965.— Вып. 4.— С. 10—22.
21. Трубин В. А. Эффективный алгоритм размещения на сети в форме дерева // Докл. АН СССР.— 1976.— Т. 231, № 3.— С. 547—550.
22. Гимади Э. Х. Эффективный алгоритм решения задачи размещения с областями обслуживания, связными относительно ациклической сети // Управляемые системы.— Новосибирск, 1983.— Вып. 23.— С. 12—23.
23. Гимади Э. Х. Задача размещения на сети с центрально-связными областями обслуживания // Управляемые системы.— Новосибирск, 1984.— Вып. 25.— С. 33—47.
24. Береснев В. Л., Гимади Э. Х., Дементьев В. Т. Экстремальные задачи стандартизации.— Новосибирск: Наука, 1978.— 333 с.
25. Гимадуддинов Э. Х. О свойствах решений одной задачи оптимального размещения точек на отрезке // Управляемые системы.— Новосибирск, 1969.— Вып. 2.— С. 77—91.
26. Гимади Э. Х. Выбор оптимальных шкал в одном классе задачи типа размещения, унификации и стандартизации // Управляемые системы.— Новосибирск, 1970.— Вып. 6.— С. 57—70.
27. Гимади Э. Х., Дементьев В. Т. Некоторые задачи выбора оптимальных параметрических рядов и методы их решения (задачи стандартизации) // Проблемы кибернетики.— 1973.— Вып. 27.— С. 19—32.
28. Береснев В. Л. Алгоритмы минимизации полиномов от булевых переменных // Проблемы кибернетики.— 1979.— Вып. 36.— С. 225—246.
29. Гимади Э. Х. Задача стандартизации с данными произвольного знака и связными, квазивыпуклыми и почти квазивыпуклыми матрицами // Управляемые системы.— Новосибирск, 1987.— Вып. 27.— С. 3—11.
30. Белинская И. Г. Об одном классе полиномов от булевых переменных // Управляемые системы.— Новосибирск, 1981.— Вып. 21.— С. 6—12.
31. Гимади Э. Х. Обоснование априорных оценок качества приближенного решения задачи стандартизации // Управляемые системы.— Новосибирск, 1987.— Вып. 27.— С. 12—27.