

## Задача размещения производства

**Дано:**  $I = \{1, \dots, m\}$  — множество районов (городов, областей), где можно открыть производство некоторой продукции;

$J = \{1, \dots, n\}$  — множество потребителей этой продукции (клиентов);

$f_i \geq 0$  — затраты на организацию производства в пункте  $i$ .

$c_{ij} \geq 0$  — производственно–транспортные расходы на удовлетворение спроса  $j$ -го клиента  $i$ -м предприятием.

$p > 0$  — максимально возможное число предприятий.



**Найти:** подмножество предприятий  $S \subset I$ ,  $|S| \leq p$ , которое позволяет удовлетворить спрос всех клиентов с минимальными суммарными затратами.

## Переменные задачи:

$$x_i = \begin{cases} 1, & \text{если открываем предприятие } i, \\ 0 & \text{в противном случае} \end{cases}$$

$$z_{ij} = \begin{cases} 1, & \text{если предприятие } i \text{ обслуживает клиента } j, \\ 0 & \text{в противном случае} \end{cases}$$

## Математическая модель

$$\min \left\{ \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} + \sum_{i \in I} f_i x_i \right\}$$

при ограничениях:

$$\sum_{i \in I} z_{ij} = 1, \quad j \in J;$$

$$x_i \geq z_{ij}, \quad i \in I, \quad j \in J;$$

$$\sum_{i \in I} x_i \leq p;$$

$$z_{ij}, x_i \in \{0, 1\}, \quad i \in I, j \in J.$$

**Упражнение.** Замена  $z_{ij} \in \{0, 1\}$  на  $z_{ij} \geq 0$  не меняет минимума суммарных затрат.

# Задача размещения с предпочтениями клиентов

**Дано:**  $I = \{1, \dots, m\}$  — множество мест, где можно открыть производство некоторой продукции;  
 $J = \{1, \dots, n\}$  — множество потребителей этой продукции (клиентов);  
 $f_i \geq 0$  — затраты на организацию производства в пункте  $i$ .  
 $c_{ij} \geq 0$  — производственно–транспортные расходы на удовлетворение спроса  $j$ -го клиента  $i$ -м предприятием.  
 $p > 0$  — максимально возможное число предприятий.



$d_{ij} \geq 0$  — предпочтения  $j$ -го клиента на множестве предприятий:

$\min_{i \in I} d_{ij}$  — наиболее желаемый поставщик;

$\max_{i \in I} d_{ij}$  — наименее желаемый поставщик.

**Найти:** Подмножество  $S \subset I$ ,  $|S| \leq p$ , открываемых предприятий, которые позволили бы удовлетворить спрос всех клиентов с минимальными суммарными затратами.

### Переменные задачи:

$$x_i = \begin{cases} 1, & \text{если открываем предприятие } i, \\ 0 & \text{в противном случае} \end{cases}$$

$$z_{ij} = \begin{cases} 1, & \text{если предприятие } i \text{ обслуживает клиента } j, \\ 0 & \text{в противном случае} \end{cases}$$

## Математическая модель

$$\min \left\{ \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij}^* + \sum_{i \in I} f_i x_i \right\}$$

при ограничениях:

$$\sum_{i \in I} x_i \leq p;$$

$$x_i \in \{0,1\}, \quad i \in I,$$

где  $z_{ij}^*$  — оптимальное решение задачи потребителя:

$$\min \sum_{j \in J} \sum_{i \in I} d_{ij} z_{ij}$$

при ограничениях:

$$\sum_{i \in I} z_{ij} = 1, \quad j \in J;$$

$$x_i \geq z_{ij}, \quad i \in I, \quad j \in J;$$

$$z_{ij} \in \{0,1\}, \quad i \in I, \quad j \in J.$$

# Квадратичная задача о назначениях

**Дано:**  $I = \{1, \dots, n\}$  — множество зданий, где могут размещаться цеха;

$J = \{1, \dots, n\}$  — множество производственных цехов;

$a_{kl}$  — расстояние между зданиями  $k, l \in I$ ;

$b_{ij}$  — объем продукции, транспортируемый из цеха  $i$  в цех  $j$ , где  $i, j \in J$ .



**Найти:** Размещение цехов по зданиям, при котором суммарный объем перевозок продукции был бы минимальным.

## Переменные задачи:

$$x_{ik} = \begin{cases} 1, & \text{если цех } i \text{ размещается в здании } k, \\ 0 & \text{в противном случае} \end{cases}$$

## Математическая модель

$$\min \sum_{i \in J} \sum_{j \in J} \sum_{l \in I} \sum_{k \in I} a_{kl} b_{ij} x_{ik} x_{jl}$$

при ограничениях:

$$\sum_{i \in J} x_{ik} = 1, \text{ для всех } k \in I;$$

$$\sum_{k \in I} x_{ik} = 1, \text{ для всех } i \in J;$$

$$x_{ik} \in \{0,1\}, \quad i \in J, k \in I.$$

# Генетический алгоритм для задач размещения

Идея заимствована у живой природы и состоит в организации эволюции, целью которой является получение оптимального решения в сложной комбинаторной задаче:

$$\min\{f(S), S \in Sol\}.$$



**Начальная популяция**  $P = \{S_1, S_2, \dots, S_k\}$  — набор допустимых решений исходной задачи.

**Шаг эволюции:** выбираем из популяции два решения, скрещиваем их, применяем мутацию, локальную перестройку и добавляем в популяцию, затем наихудшее решение удаляем из популяции.



## Общая схема алгоритма

1. Выбрать начальную популяцию  $P$  и запомнить рекорд  $F^* = \min_{i=1,\dots,k} f(S_i)$ .
2. Пока не выполнен критерий останова делать следующее:
  - 2.1. Выбрать “родителей”  $S_{i_1}, S_{i_2}$  из популяции.
  - 2.2. Применить к  $S_{i_1}, S_{i_2}$  оператор скрещивания и получить новое решение  $S'$ .
  - 2.3. Применить к  $S'$  оператор мутации и получить новое решение  $S''$ .
  - 2.4. Применить к  $S''$  оператор локального улучшения и получить новое решение  $S'''$ .
  - 2.5. Если  $f(S''') < F^*$ , то сменить рекорд  $F^* := f(S''')$ .
  - 2.6. Добавить  $S'''$  к популяции и удалить из нее наихудшее решение.

## Оператор скрещивания

Пусть  $S_1, S_2$  — два решения, задаваемые векторами  $X^1, X^2 \in \{0,1\}^n$ .

*Одноточечный оператор скрещивания:* выбираем случайным образом координату  $1 \leq l \leq n$  и новый вектор  $X'$  получает первые  $l$  координат от вектора  $X^1$ , а остальные от вектора  $X^2$ .

$$\begin{array}{l} X^1: (0 \ 1 \ 0 \ 0 \mid 1 \ . \ . \ . \ 0 \ 1 \ 1) \\ X^2: (1 \ 1 \ 0 \ 1 \mid 0 \ . \ . \ . \ 1 \ 1 \ 1) \\ X': (0 \ 1 \ 0 \ 0 \mid 0 \ . \ . \ . \ 1 \ 1 \ 1) \end{array}$$

$\underbrace{\hspace{10em}}_l$

Аналогично определяется двухточечный, трехточечный и т.д. операторы.

*Равномерный оператор скрещивания:* новое решение  $X'$  в каждой координате получает с вероятностью 0.5 значение одного из родителей.

## Выбор родителей

*Турнирная селекция:* из популяции  $P$  случайным образом выбирается некоторое подмножество  $P' \subseteq P$  и родителем назначается наилучшее решение в  $P'$ :

$$S_i = \min_{S \in P'} f(S).$$

*Пропорциональная селекция:* из популяции  $P$  случайным образом выбираются два родителя. Для решения  $S_i$  вероятность быть выбранным обратно пропорциональна значению целевой функции  $f(S_i)$ .

*Варианты:* Лучший в популяции + случайно выбранный.  
Случайно выбранный + наиболее удаленный от него и др.



## Локальное улучшение

Для решения  $S$  обозначим через  $N(S)$  его окрестность, например, множество всех решений  $S'$ , находящихся от  $S$  на расстоянии не более 2 (3, 4, 5...)

### Алгоритм локального спуска

1. Положить  $S := S''$ ;
2. Найти в окрестности решения  $S$  наилучшего соседа  $\bar{S}$

$$f(\bar{S}) = \min\{f(\tilde{S}), \tilde{S} \in N(S)\};$$

3. Если  $f(\bar{S}) < f(S)$ , то положить  $S := \bar{S}$  и вернуться на шаг 2, иначе STOP, получен локальный минимум.

## Задача размещения в условиях конкуренции

$I = \{1, \dots, m\}$  — множество пунктов размещения;

$J = \{1, \dots, n\}$  — множество клиентов;

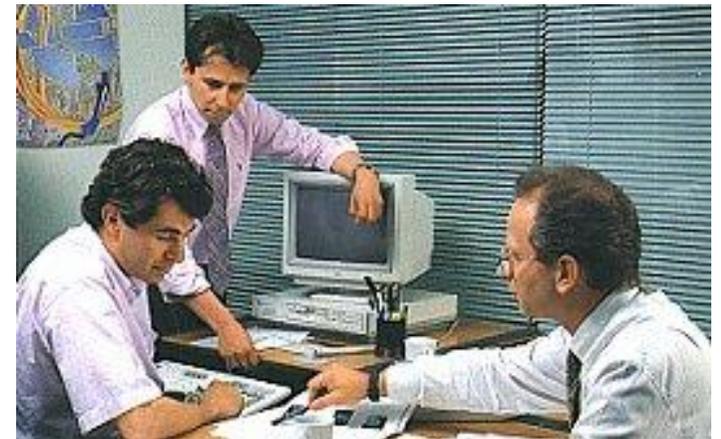
$c_{ij}$  — расстояние от пункта  $i$  до клиента  $j$ ;

$p$  — число предприятий, открываемых ЛПР<sub>1</sub>;

$r$  — число предприятий, открываемых ЛПР<sub>2</sub>;

$d_j$  — доход от обслуживания клиента  $j$ ;

Сначала ЛПР<sub>1</sub> принимает решение об открытии своих предприятий. Затем, зная это решение, ЛПР<sub>2</sub> открывает свои предприятия так, чтобы получить максимальный доход. Клиент знает оба решения и выбирает ближайшее к себе предприятие. Задача состоит в том, чтобы найти решение для ЛПР<sub>1</sub> с максимальным доходом.



## Переменные задачи

$$x_i = \begin{cases} 1, & \text{если ЛПР}_1 \text{ открыл предприятие в пункте } i \\ 0 & \text{в противном случае} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{если ЛПР}_2 \text{ открыл предприятие в пункте } i \\ 0 & \text{в противном случае} \end{cases}$$

$$z_j = \begin{cases} 1, & \text{если клиент } j \text{ обслуживается ЛПР}_1 \\ 0, & \text{если клиент } j \text{ обслуживается ЛПР}_2 \end{cases}$$

Для вектора  $x$  положим

$I_j(x) = \{i \in I \mid c_{ij} < \min(c_{kj} \mid x_k = 1)\}$  — множество пунктов размещения, которые находятся ближе к клиенту  $j$ , чем ближайшее открытое предприятие ЛПР<sub>1</sub>.

# Математическая модель двухуровневого программирования

$$\max_x \sum_{j \in J} d_j z_j^*(x)$$

при ограничениях 
$$\sum_{i \in I} x_i = p, \quad x_i \in \{0,1\}, \quad i \in I$$

где  $z_j^*$  — оптимальное решение задачи ЛПР<sub>2</sub>:

$$\max_{z,y} \sum_{j \in J} d_j (1 - z_j)$$

при ограничениях 
$$1 - z_j \leq \sum_{i \in I_j(x)} y_i, \quad j \in J$$

$$\sum_{i \in I} y_i = r$$

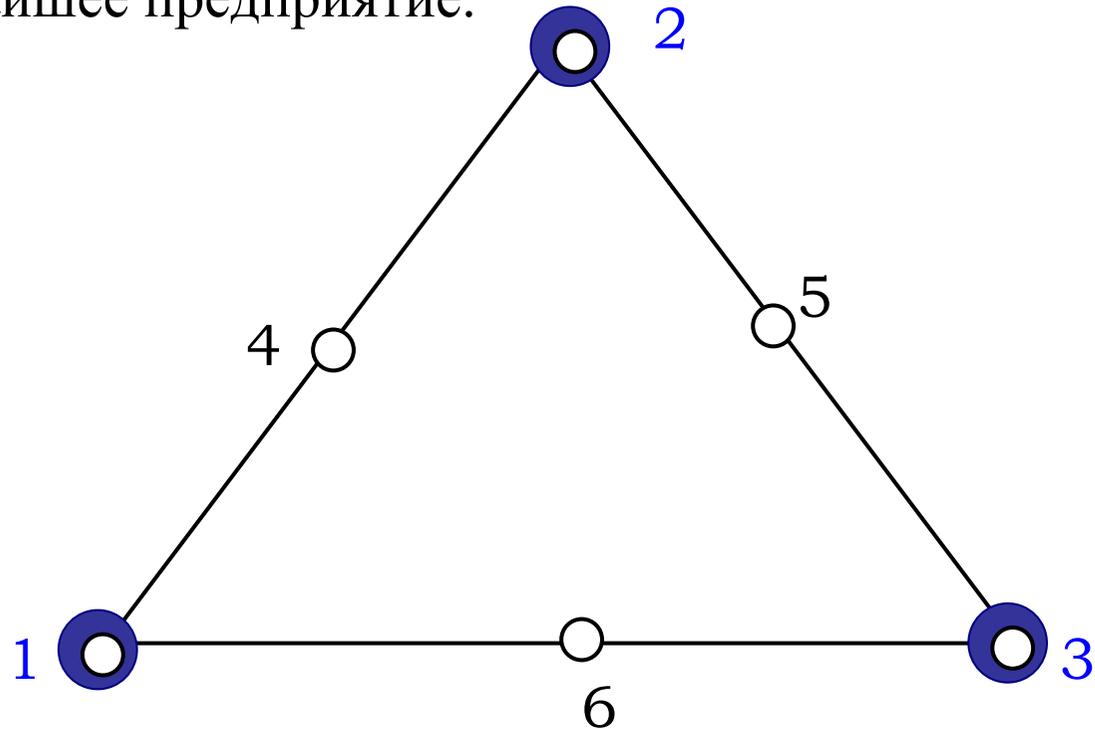
$$x_i + y_i \leq 1, \quad i \in I, \quad y_i, z_j \in \{0,1\}$$

## «Безнадежный» пример

$I = \{1, 2, 3, 4, 5, 6\}$  — места размещения предприятий;

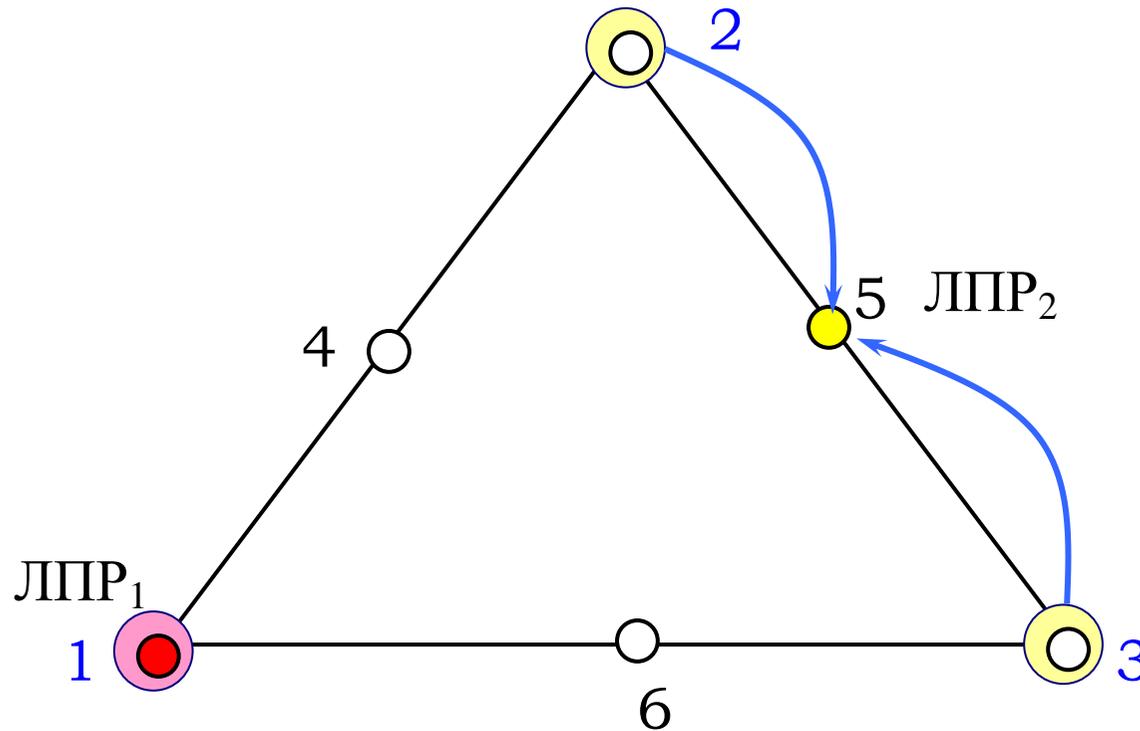
$J = \{1, 2, 3\}$  — клиенты.

Клиенты выбирают ближайшее предприятие.

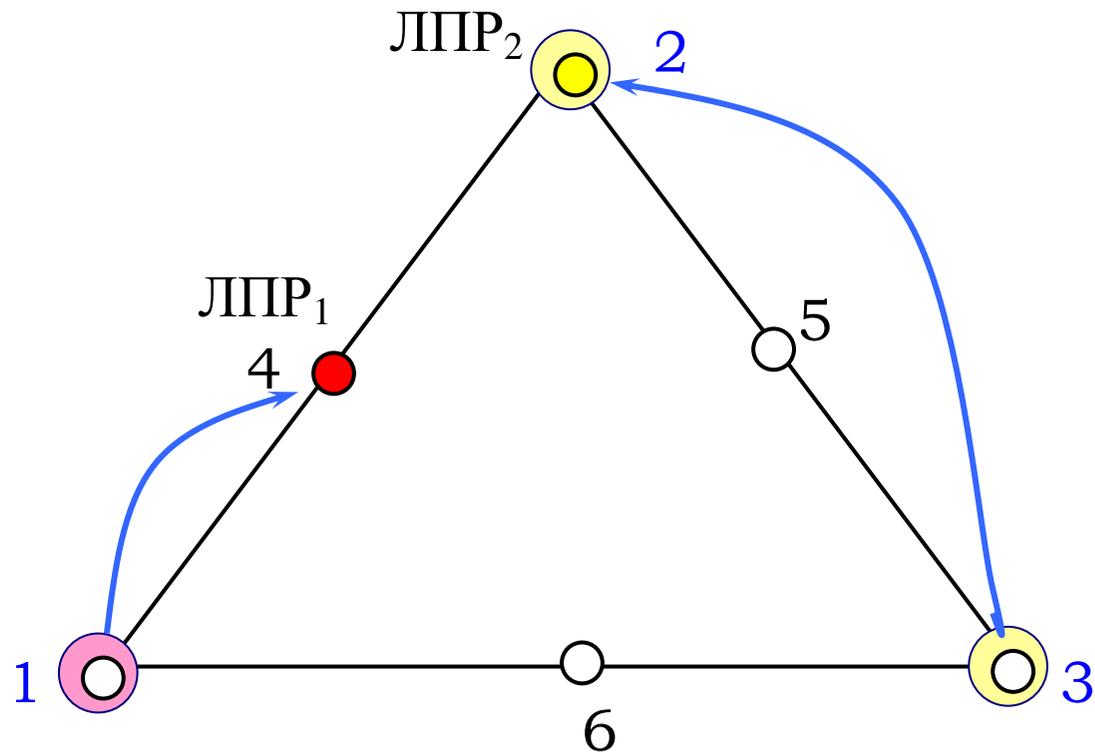


При  $p = r = 1$  ЛПР<sub>1</sub> всегда проигрывает!

Если ЛПР<sub>1</sub> ставит предприятие в вершине треугольника, то ЛПР<sub>2</sub> ставит свое предприятие на противоположной стороне треугольника и захватывает двух клиентов, в то время как ЛПР<sub>1</sub> получает только одного.



Если ЛПР<sub>1</sub> ставит предприятие на стороне треугольника, то ЛПР<sub>2</sub> ставит свое предприятие в соседней вершине треугольника и тоже захватывает двух клиентов, в то время как ЛПР<sub>1</sub> получает только одного



## Численные методы

При заданном решении  $x$  для ЛПР<sub>1</sub> задача ЛПР<sub>2</sub> является NP-трудной задачей целочисленного линейного программирования.

Зная ее оптимальное решение, можно вычислить доход ЛПР<sub>2</sub> и ЛПР<sub>1</sub>.

- Генетический локальный поиск в пространстве переменных  $x$  для ЛПР<sub>1</sub>.
- Имитация отжига в пространстве переменных  $x$  для ЛПР<sub>1</sub>.
- Поиск с чередующимися окрестностями в пространстве переменных  $x$  для ЛПР<sub>1</sub>.

## Точный метод

Пусть  $\mathcal{F}$  — непустое семейство решений ЛПР<sub>2</sub>.

Для  $y \in \mathcal{F}$  положим

$$I_j(y) = \{i \in I \mid c_{ij} \leq \min_{l \in I} c_{lj} \mid y_l = 1\}$$

Это множество предприятий, позволяющих ЛПР<sub>1</sub> удержать клиента  $j$ , если ЛПР<sub>2</sub> использует решение  $y$ .

Дополнительные переменные:

$D$  — доход ЛПР<sub>1</sub>

$$z_{ij} = \begin{cases} 1, & \text{если предприятие } i \text{ ЛПР}_1 \text{ ближайшее для клиента } j \\ 0 & \text{в противном случае} \end{cases}$$

## Переформулировка задачи:

$$\max D \tag{1}$$

при ограничениях 
$$\sum_{j \in J} \sum_{i \in I_j(y)} d_j z_{ij} \geq D, \quad \forall y \in \mathcal{F}, \tag{2}$$

$$\sum_{i \in I} z_{ij} = 1, \quad j \in J, \tag{3}$$

$$x_i \geq z_{ij}, \quad i \in I, j \in J, \tag{4}$$

$$\sum_{i \in I} x_i = p. \tag{5}$$

Если  $\mathcal{F}$  — все решения ЛПР<sub>2</sub>, то получаем эквивалентную переформулировку.

Пусть  $\mathcal{F}$  — некоторое семейство решений ЛПР<sub>2</sub>.

Тогда  $D(\mathcal{F})$  — верхняя оценка максимального дохода ЛПР<sub>1</sub>.

Пусть  $x(\mathcal{F})$  — оптимальное решение для заданного  $\mathcal{F}$  и  $D'(x(\mathcal{F}))$  — доход ЛПР<sub>1</sub> на решении  $x(\mathcal{F})$ .  $D'(x(\mathcal{F}))$  — нижняя оценка оптимума для ЛПР<sub>1</sub>.

### Итерационный метод:

0. Выбрать начальное семейство  $\mathcal{F}$  и положить  $D^* := 0$
1. Решить задачу (1)–(5), найти  $D(\mathcal{F})$  и  $x(\mathcal{F})$
2. Решить задачу ЛПР<sub>2</sub>, вычислить  $y(x(\mathcal{F}))$  и  $D'(x(\mathcal{F}))$
3. Если  $D^* < D'(x(\mathcal{F}))$ , то  $D^* := D'(x(\mathcal{F}))$
4. Если  $D(\mathcal{F}) = D^*$ , то STOP
5. Добавить  $y(x(\mathcal{F}))$  в семейство  $\mathcal{F}$  и вернуться на шаг 1.

**Теорема.** Итерационный метод является конечным и дает точное решение задачи (1)–(5).

**Доказательство.** Метод останавливается, когда верхняя оценка оптимума совпадает с нижней оценкой. Значит решение точное. Так как возможных решений ЛПР<sub>2</sub> не более  $C_n^r$ , то для конечности метода требуется, чтобы решения  $y(x(\mathcal{F}))$  на шаге 5 не повторялись.

Предположим противное. Пусть решение  $y(x(\mathcal{F}))$  уже содержится в  $\mathcal{F}$ . Тогда согласно ограничению (2) верхняя оценка  $D(\mathcal{F})$  не должна превосходить дохода ЛПР<sub>1</sub> при ответном ходе  $y(x(\mathcal{F}))$  ЛПР<sub>2</sub>. Но  $y(x(\mathcal{F}))$  — оптимальное решение ЛПР<sub>2</sub> для  $x(\mathcal{F})$ . Значит  $D(\mathcal{F})$  не превосходит нижней оценки  $D'(x(\mathcal{F}))$ , и метод должен был остановиться на шаге 4, не попадая на шаг 5. Противоречие. ■

## Доля рынка ЛПР<sub>1</sub>

