

# Теория принятия решений



# Теория принятия решений

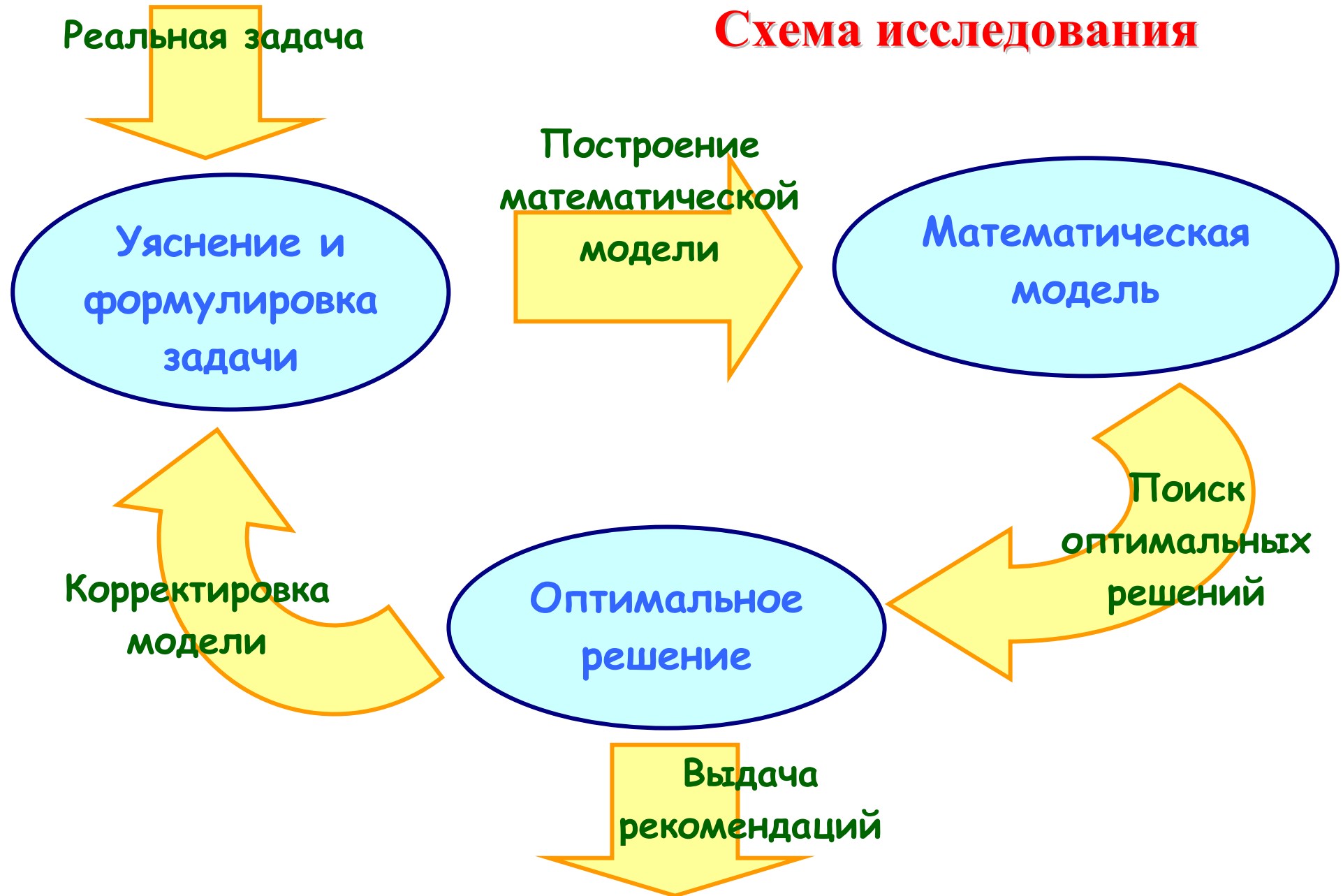
*Исследование операций* — теория математических моделей и методов принятия решений.

1. Наличие некоторого **процесса**
2. Наличие **управляющих воздействий**
3. Наличие **цели**, ради которой проводится операция
4. **Выбор наилучшего (оптимального) управления**, при котором достигается цель

*Операция* — система действий, объединенная единым замыслом и направленная на достижение определенной цели.

Основная задача теории оптимальных решений состоит в представлении обоснованных количественных данных и рекомендаций для принятия оптимальных решений.

## Схема исследования



# Математическая модель

*Математическая модель* — объективная схематизация основных аспектов решаемой задачи или ее описание в математических терминах.

Математическая модель описывает исследуемую систему и позволяет выразить ее эффективность в виде *целевой функции*

$$W = f(X, Y),$$

где  $X = (x_1, \dots, x_n)$  — управляемые переменные,

$Y = (y_1, \dots, y_m)$  — неуправляемые переменные (исходные данные).

Связь между переменными  $X$  и исходными данными  $Y$  выражается с помощью ограничений

$$\varphi(X, Y) \leq 0.$$

# Модели принятия решений

1. Долгосрочное стратегическое планирование:

задачи размещения производства, развитие нефтяной и газовой промышленности

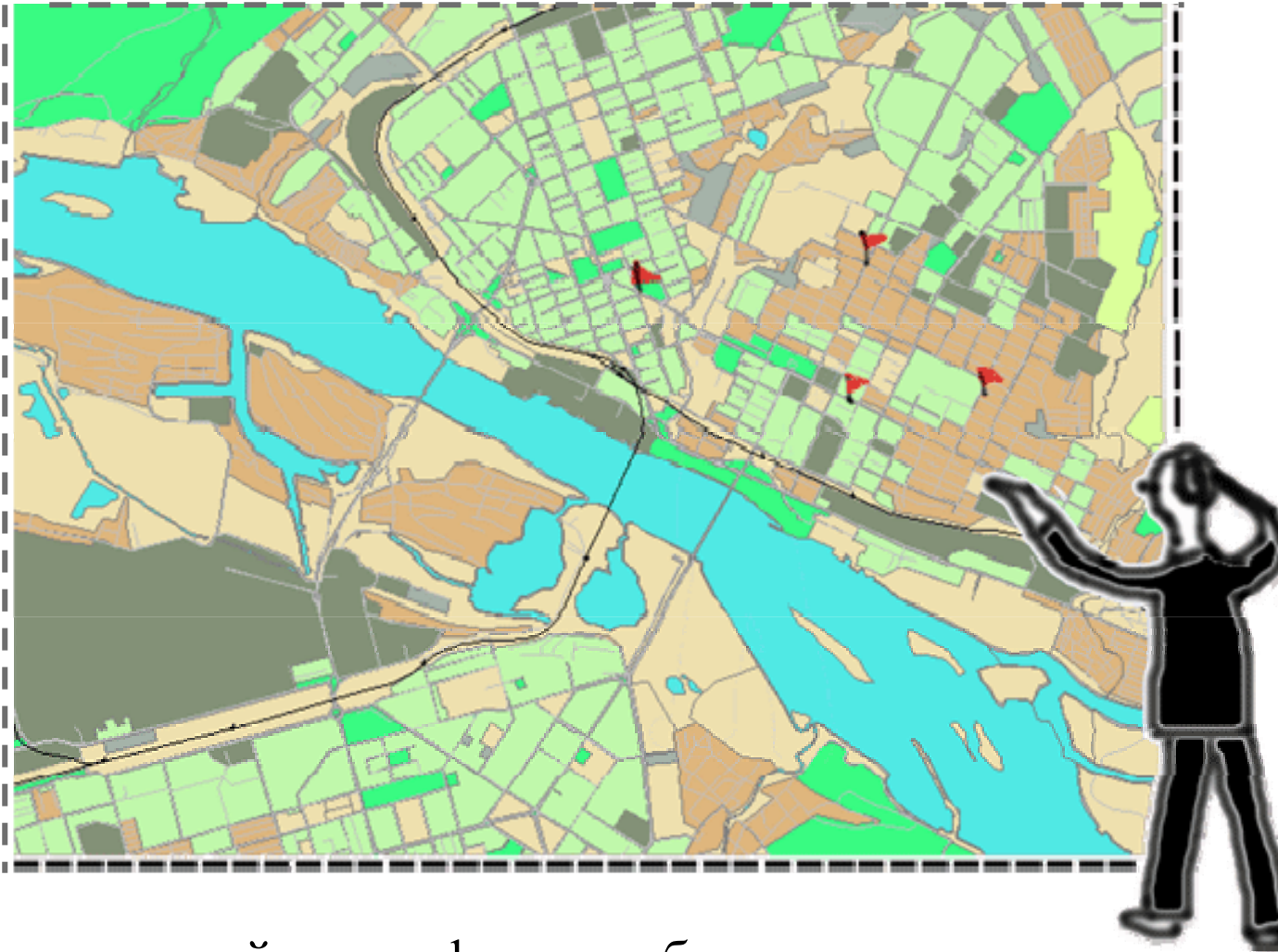
2. Среднесрочное планирование:

транспортные задачи, задачи маршрутизации, задачи календарного планирования с ограниченными ресурсами

3. Оперативное управление:

задачи теории расписаний, задачи раскроя и упаковки

# Задачи размещения производства



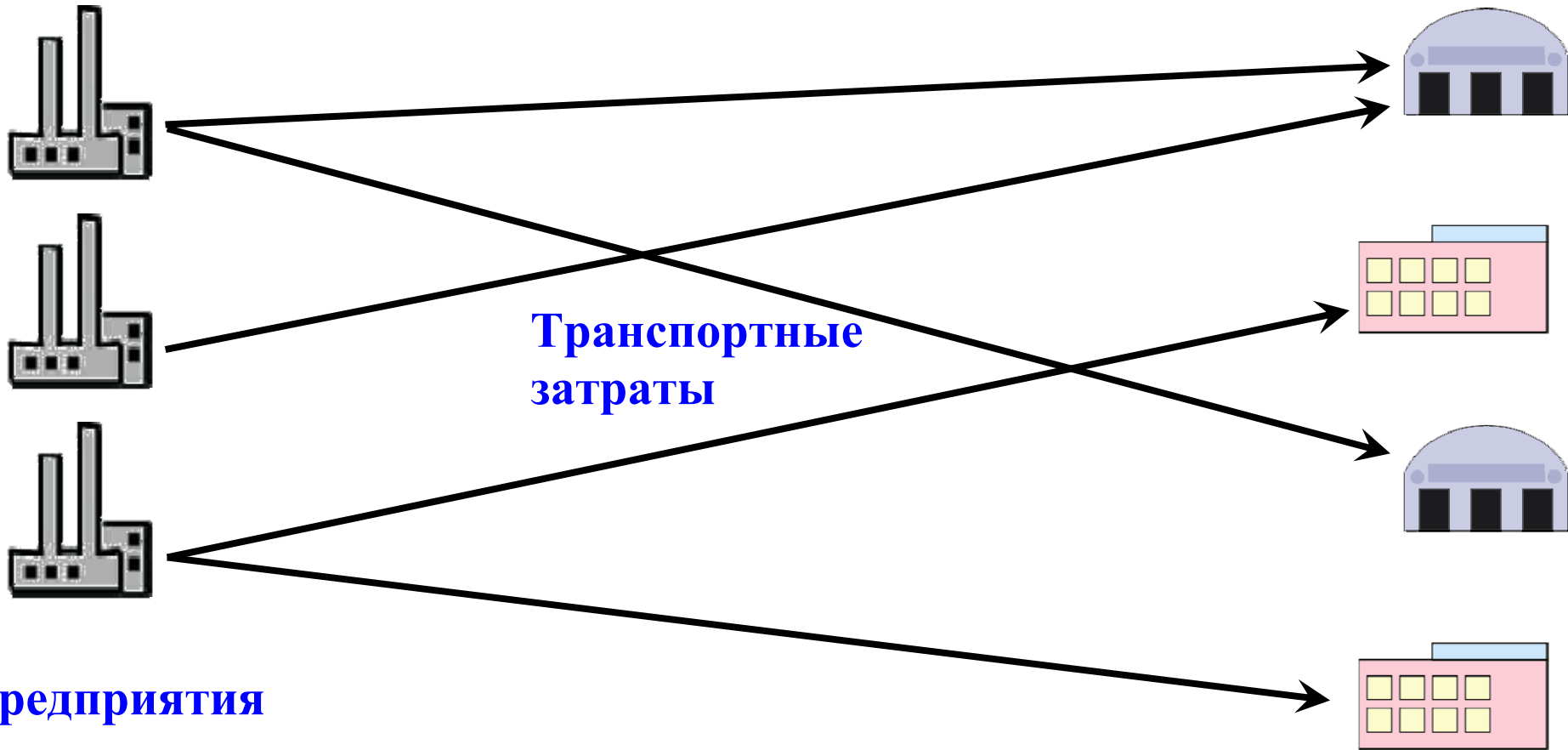
Системы сотовой связи, филиалы банков, производство продукции

---

*Лекция 1. Исследование операций. Динамическое программирование*

## Транспортные задачи

Потребители



Предприятия

Минимизировать затраты на перевозку продукции

## Задачи маршрутизации



Найти маршрут минимальной длины



# Задачи теории расписаний

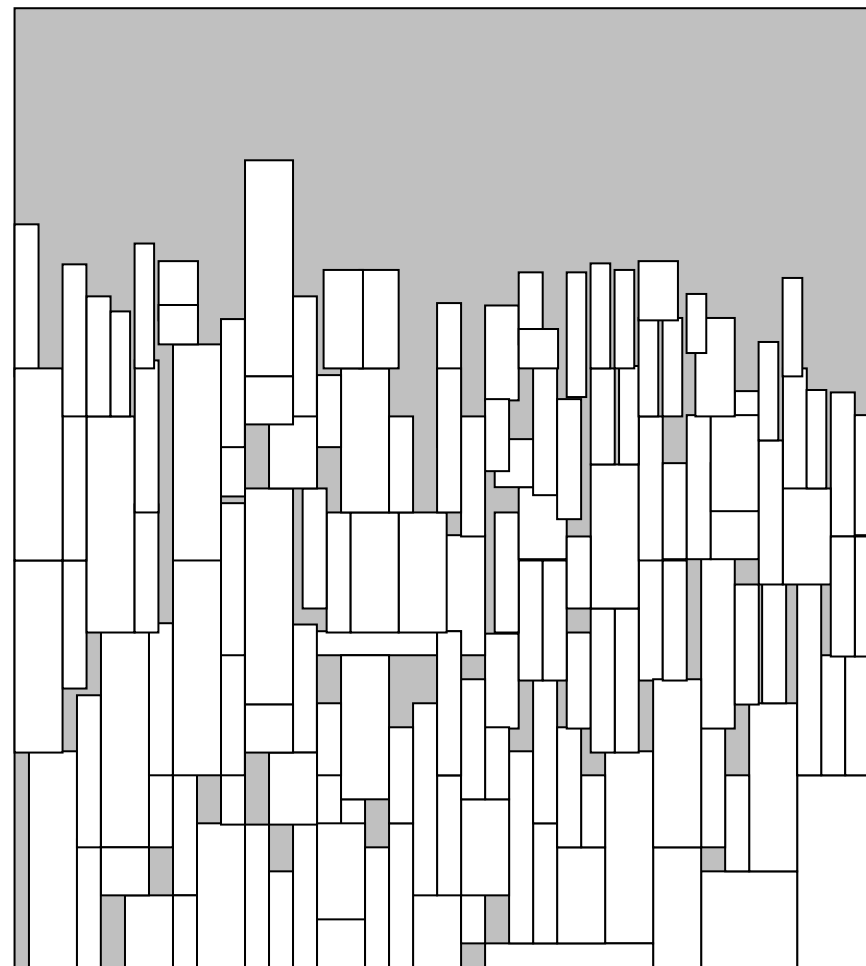
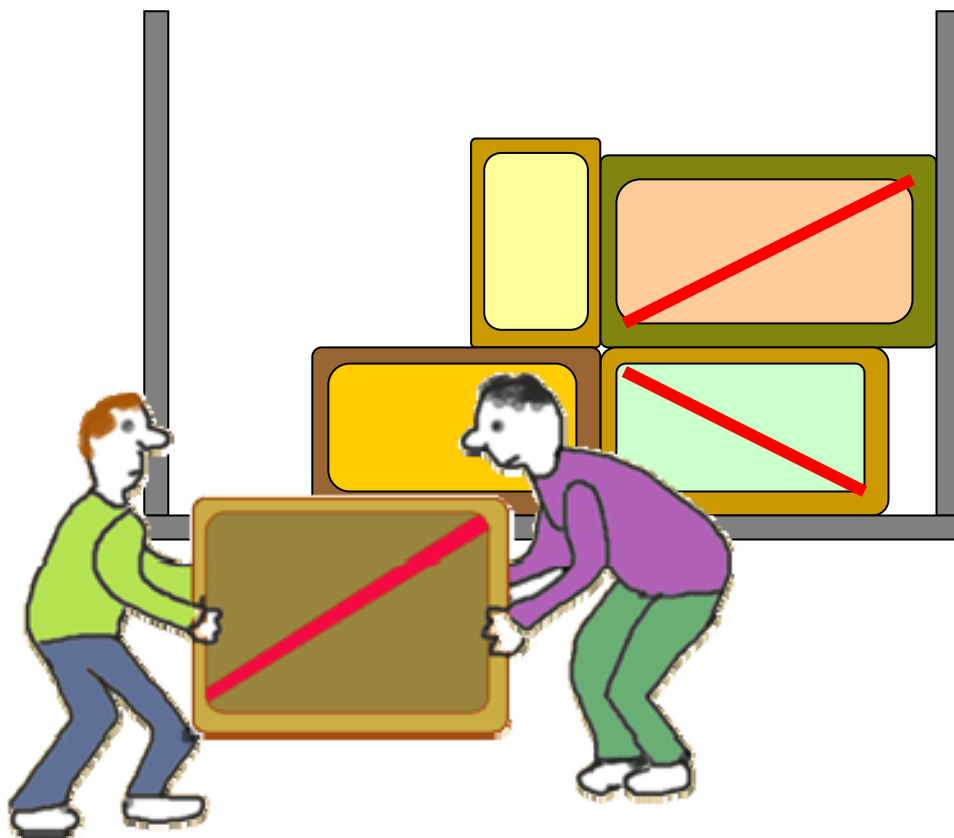
14:25

007	Москва – Владивосток	02:30	02:50	007	Москва – Владивосток	02:30	02:50
874	Москва – Одесса	11:05	11:25	874	Москва – Одесса	11:05	11:25
65	Урюпинск – Киев	12:20	12:45	65	Урюпинск – Киев	12:20	12:45
874	Новосибирск – Бийск	14:45	14:55	874	Новосибирск – Бийск	14:45	14:55
007	Барнаул – Магистрала	16:00	16:10	007	Барнаул – Магистрала	16:00	16:10
874	Москва – Карелия	18:25	18:45	874	Москва – Карелия	18:25	18:45



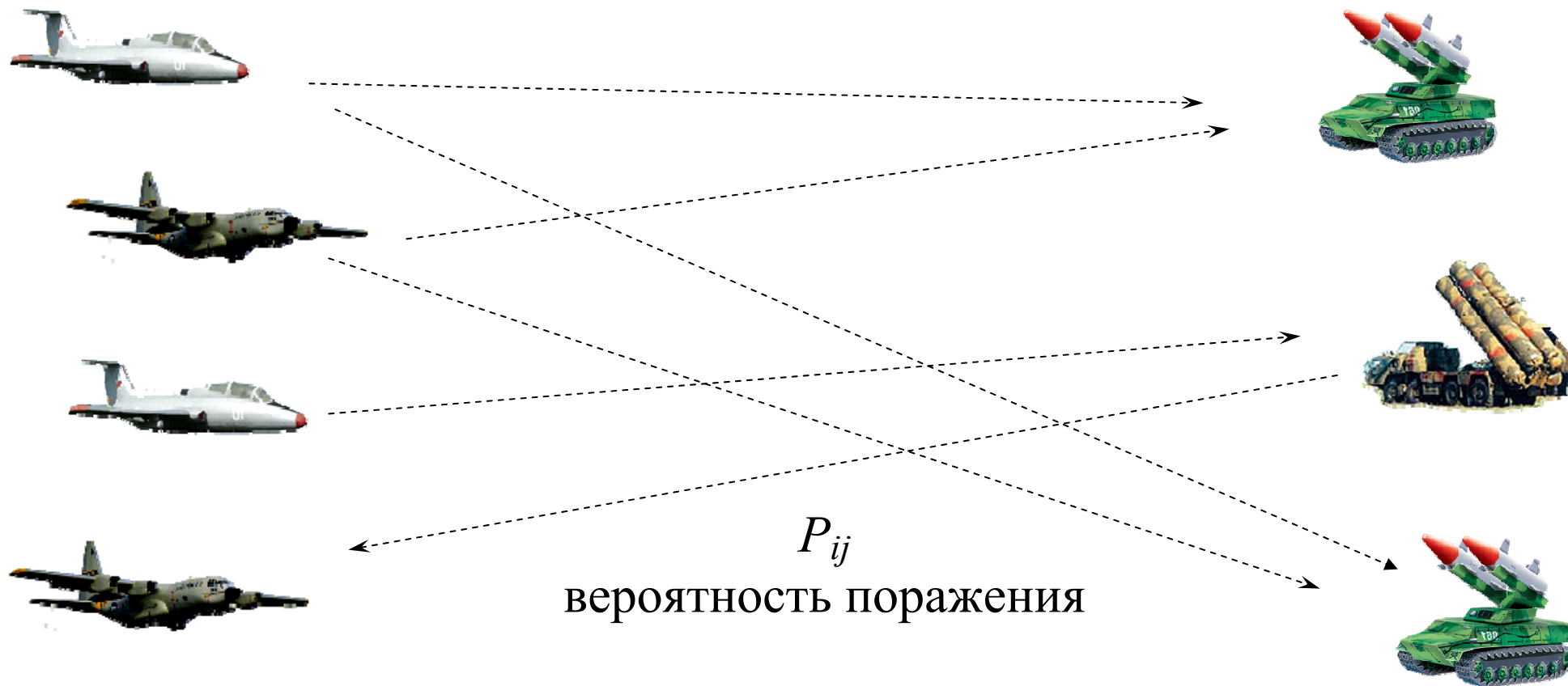
Графики движения поездов, рабочие бригады, ремонт составов

## Задачи раскроя и упаковки

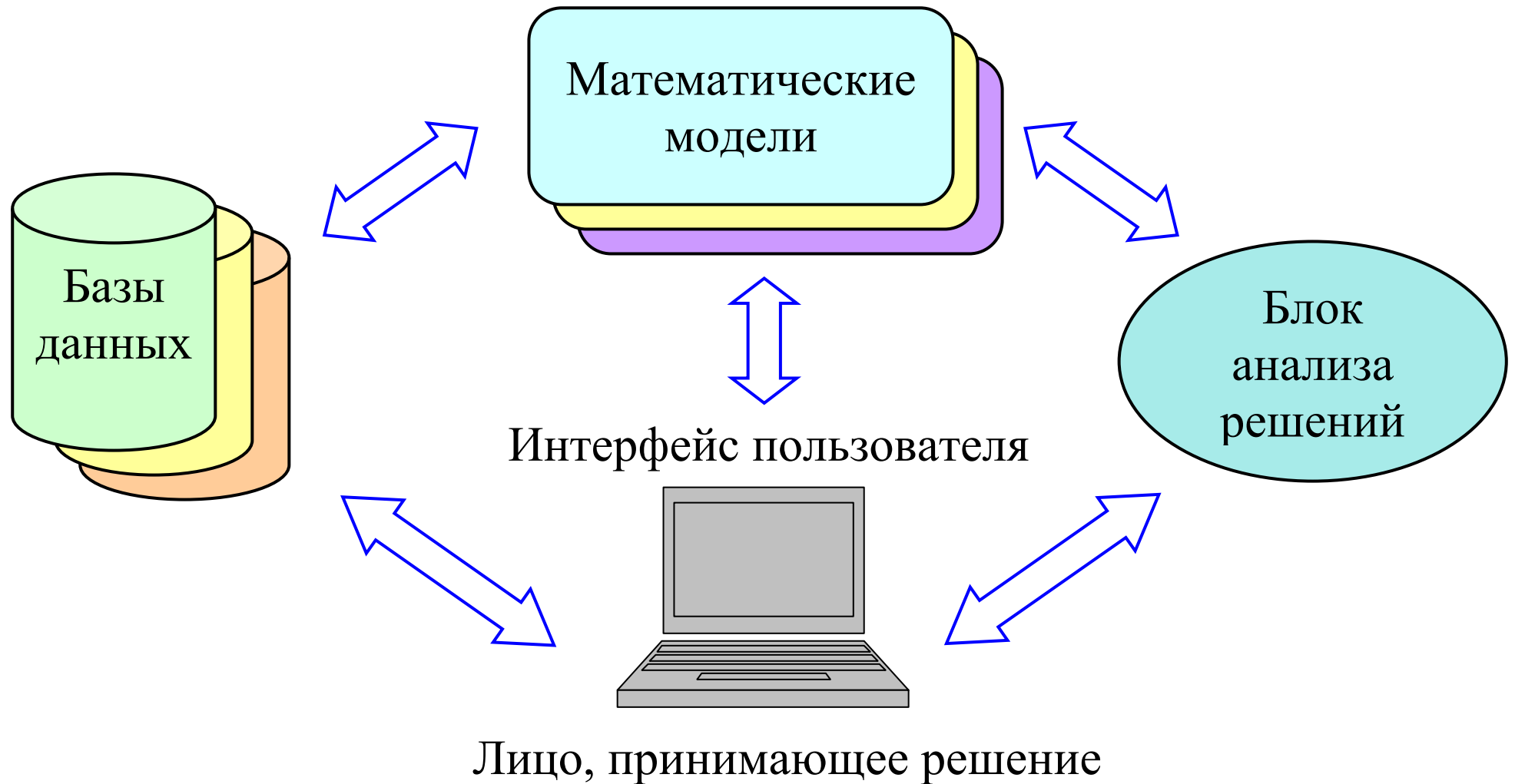


Раскрой пиломатериала, листового железа, станки с ЧПУ

# Матричные игры



# Системы поддержки решений



# Характеристики алгоритмов

Для оценки качества алгоритмов будем использовать два параметра:

$T_A$  — *трудоемкость* (число элементарных операций алгоритма  $A$ );

$P_A$  — требуемый *объем памяти*.

*Элементарная операция* — одна из арифметических операций: сложение, вычитание, умножение, деление или логическая операция сравнение двух чисел.

Нас будет интересовать зависимость параметров алгоритма от длины записи исходных данных задачи с точностью до порядка величин.

**Пример:** При  $T = 3/2 n^2$ , будем писать  $T = O(n^2)$  или  $T \approx n^2$ .

# Полиномиальные алгоритмы

**Определение.** Алгоритм  $A$  называют *полиномиальным*, если его трудоемкость  $T_A$  ограничена полиномом от длины записи исходных данных, то есть существует константа  $c > 0$  и натуральное число  $k$  такие, что  $T_A \leq cL^k$ , где  $L$  — длина записи исходных данных.

**Пример:** Пусть  $f_i(x_i) = a_i x_i$ , тогда 
$$L = \sum_{i=1}^n \log a_i + \log Y,$$

но  $T_{\text{ДП}} = O(Y^2 n)$ , то есть алгоритм ДП не является полиномиальным.

# Распределительная задача

Имеем

$n$  — число предприятий;

$Y$  — количество единиц некоторого ресурса;

$f_k(x)$  — количество продукции, которое будет произведено на  $k$ -м предприятии, если в него будет вложено  $x$  единиц ресурса (монотонно неубывающая функция).

Требуется: максимизировать объем продукции

$$f_1(x_1) + \dots + f_n(x_n) \rightarrow \max \quad (1)$$

$$x_1 + \dots + x_n \leq Y \quad (2)$$

$$x_i \geq 0, \text{ целые, } i = 1, \dots, n. \quad (3)$$

# Идея динамического программирования (ДП)

Метод ДП (Р. Беллман, В.С. Михалевич, Н.З. Шор ) можно трактовать как алгоритмическую версию рассуждений по индукции.

Пусть  $s_k(y)$ ,  $1 \leq k \leq n$ ,  $0 \leq y \leq Y$ , — оптимальное значение целевой функции задачи (1) – (3), где  $n$  заменено на  $k$ ,  $Y$  заменено на  $y$ .

Требуется найти  $s_n(Y)$  и набор переменных, на котором достигается это значение.



**ТЕОРЕМА 1.** Пусть  $f_1, \dots, f_n$  — монотонно неубывающие функции. Тогда справедливы следующие *рекуррентные соотношения*:

$$s_1(y) = f_1(y), \quad 0 \leq y \leq Y; \quad (4)$$

$$s_k(y) = \max \{s_{k-1}(y - x) + f_k(x) \mid 0 \leq x \leq y\}, \quad 2 \leq k \leq n, \quad 0 \leq y \leq Y, \quad (5)$$

**Доказательство:** Соотношение (4) очевидно. По определению

$$s_k(y) \geq \max \{s_{k-1}(y - x) + f_k(x) \mid 0 \leq x \leq y\}.$$

Пусть теперь  $(x_1^*, \dots, x_k^*)$  — такой вектор, что  $x_1^* + \dots + x_k^* \leq y$  и

$$s_k(y) = f_1(x_1^*) + \dots + f_k(x_k^*).$$

Поскольку  $s_{k-1}(y - x_k^*) \geq f_1(x_1^*) + \dots + f_{k-1}(x_{k-1}^*)$ , имеем

$$s_k(y) = f_1(x_1^*) + \dots + f_k(x_k^*) \leq s_{k-1}(y - x_k^*) + f_k(x_k^*). \quad \blacksquare$$

Алгоритм ДП вычисляет множество  $S_k = \{s_k(y) \mid 0 \leq y \leq Y\}$ ,  $k = 1, \dots, n$  с помощью соотношений (4) и (5), где на каждом шаге оптимизируется ровно одна переменная.

Процесс вычисления  $S_1, \dots, S_n$  называется *прямым ходом* алгоритма.

Число операций  $\approx Y^2 n$

Память  $\approx Y n$ .

$y$	$S_1(y)$	$S_2(y)$	$\dots$	$S_n(y)$
0				
1				
2				
$\vdots$				
$Y$				$S_n(Y)$

При *обратном ходе* алгоритма вычисляются значения  $(x_n^*, \dots, x_1^*)$ , с учетом того, что уже известны  $S_k(y)$ . Например,  $x_n^*$  определяется из уравнения  $s_n(Y) = f_n(x_n^*) + s_{n-1}(Y - x_n^*)$  и так далее.

Число операций  $\approx Y n$ . Память  $\approx Y n$ .

Обобщим задачу (1)–(3):

$$f_1(x_1) + \dots + f_n(x_n) \rightarrow \max \quad (1')$$

$$h_1(x_1) + \dots + h_n(x_n) \leq Y \quad (2')$$

$$a_i \geq x_i \geq 0, \text{ целые, } i = 1, \dots, n. \quad (3')$$

Если  $h_i(x)$  — целочисленные монотонно неубывающие функции, то вместо (4)–(5) можно использовать следующие

*рекуррентные соотношения:*

$$s_1(y) = f_1(x^*), \text{ где } x^* = \max \{ x \mid x \leq a_1 \mid h_1(x) \leq y \}, 0 \leq y \leq Y; \quad (4')$$

$$s_k(y) = \max_{\{x \leq a_k \mid h_k(x) \leq y\}} \{ f_k(x) + s_{k-1}(y - h_k(x)) \}, 2 \leq k \leq n, 0 \leq y \leq Y. \quad (5')$$

**Упражнение 1.** Доказать справедливость соотношений (4')–(5').

**Обратная задача** — поиск наименьших затрат на получение заданного количества продукции:

$$h_1(x_1) + \dots + h_n(x_n) \rightarrow \min \quad (6)$$

$$f_1(x_1) + \dots + f_n(x_n) \geq D \quad (7)$$

$$a_i \geq x_i \geq 0, \text{ целые, } i = 1, \dots, n. \quad (8)$$

Если  $f_k(x)$  — целочисленные монотонно неубывающие функции, то для решения задачи (6)–(8) можно использовать идеи динамического программирования.

Пусть  $f_i^{-1}(d) = \min \{0 \leq x \leq a_i \mid f_i(x) \geq d\}$ .

Для  $1 \leq k \leq n$ ,  $0 \leq d \leq D$  обозначим через  $t_k(d)$  — оптимальное решение задачи (6)–(8), в которой  $n$  заменено на  $k$ , а  $D$  заменено на  $d$ .

Требуется найти  $t_n(D)$ .

### *Рекуррентные соотношения*

$$t_1(d) = \begin{cases} \infty, & \text{если } f_1(a_1) < d, \\ h_1(f_1^{-1}(d)), & \text{если } f_1(a_1) \geq d, \end{cases} \quad 0 \leq d \leq D, \quad (9)$$

$$t_k(d) = \min \{t_{k-1}(d - f_k(x)) + h_k(x) \mid 0 \leq x \leq a_k, x \leq f_k^{-1}(d)\}, \quad (10)$$
$$k \geq 2, \quad 0 \leq d \leq D.$$

**Упражнение 2.** Доказать справедливость соотношений (9)–(10).

**ТЕОРЕМА 2:** Предположим, что  $D$  — наибольшее число, для которого оптимальное значение целевой функции задачи (6)–(8) не превосходит  $Y$ . Тогда оптимальное значение целевой функции задачи (1')–(3') равно  $D$ .

**Доказательство:** Пусть  $D$  удовлетворяет условию теоремы и  $(x_1^*, \dots, x_n^*)$  — соответствующее решение задачи (6)–(8).

Значит

$$f_1(x_1^*) + \dots + f_n(x_n^*) \geq D \quad \text{и} \quad h_1(x_1^*) + \dots + h_n(x_n^*) \leq Y.$$

Следовательно,  $D$  не превосходит оптимального решения  $D_1$  задачи (1')–(3'). Если бы  $D_1$  было больше  $D$ , то решение задачи (6)–(8), в которой  $D$  заменено на  $D_1$ , тоже не превышало бы  $Y$ , что противоречит максимальнойности  $D$ . ■

## Задача о ближайшем соседе

**Дано:** функция  $f(x, y) \geq 0$  — затраты на обслуживание отрезка дороги от  $x$  до  $y$ ,  $0 \leq x \leq y \leq M$ ,  $x, y$  — целочисленные точки,  $n$  — число отрезков.

**Найти:** оптимальное разбиение сегмента  $[0, M]$  на  $n$  отрезков.

**Математическая модель:**

$$\min \sum_{k=1}^n f(x_{k-1}, x_k)$$

$$0 = x_0 \leq \dots \leq x_n = M$$

## Алгоритм динамического программирования

$S_k(y)$  — минимальные затраты на обслуживание  $k$  отрезков для сегмента  $[0, y]$ .

### Рекуррентные соотношения:

$$S_1(y) = f(0, y), \quad y = 1, \dots, M$$

$$S_k(y) = \min_{0 \leq x \leq y} \{S_{k-1}(x) + f(x, y)\}, \quad y = 0, \dots, M, \quad k = 2, \dots, n.$$

$$T = O(nM^2) \quad \Pi = O(nM)$$



## Оптимизация числа отрезков

Для каждого  $n = 1, \dots, M$  найти  $S_n(M)$  и выбрать наименьшее значение

$$T = O(M^3), \quad \Pi = O(M^2).$$

Модифицированный вариант

$\tilde{S}(y)$  — минимальные затраты на обслуживание сегмента  $[0, y]$ .

### Рекуррентные соотношения:

$$\tilde{S}(0) = 0,$$

$$\tilde{S}(y) = \min_{0 \leq x \leq y-1} \{\tilde{S}(x) + f(x, y)\}, \quad y = 1, \dots, M.$$

$$T = O(M^2), \quad \Pi = O(M).$$