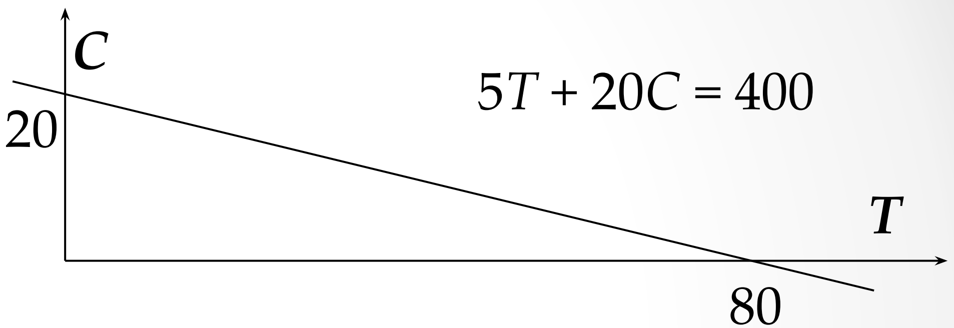
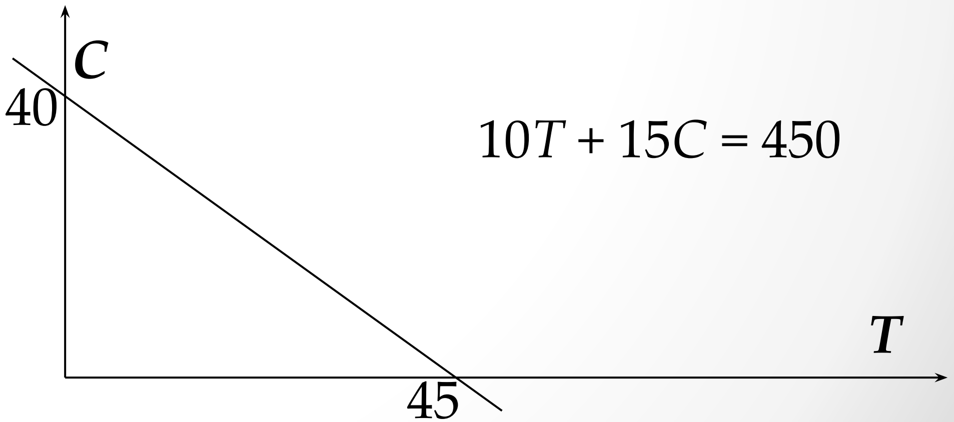
Решение задач оптимизации

Линейное программирование

(Целевая функция и функция ограничений – линейные)

Производственная задача	Стол (C)	Табурет (T)	Ограничение (R)
Материал	20	5	400
Работа	15	10	450
Прибыль	80	45	Max





Линейное программирование
(Графическое решение)

$$45T + 80C \to \mathbf{MAX}$$
$$5T + 20C \le 400$$
$$10T + 15C \le 450$$
$$C > 0;$$
$$T > 0$$

Находим точку *A*:

$5T + 20C = 400$ (1)

$10T + 15C = 450$ (2)

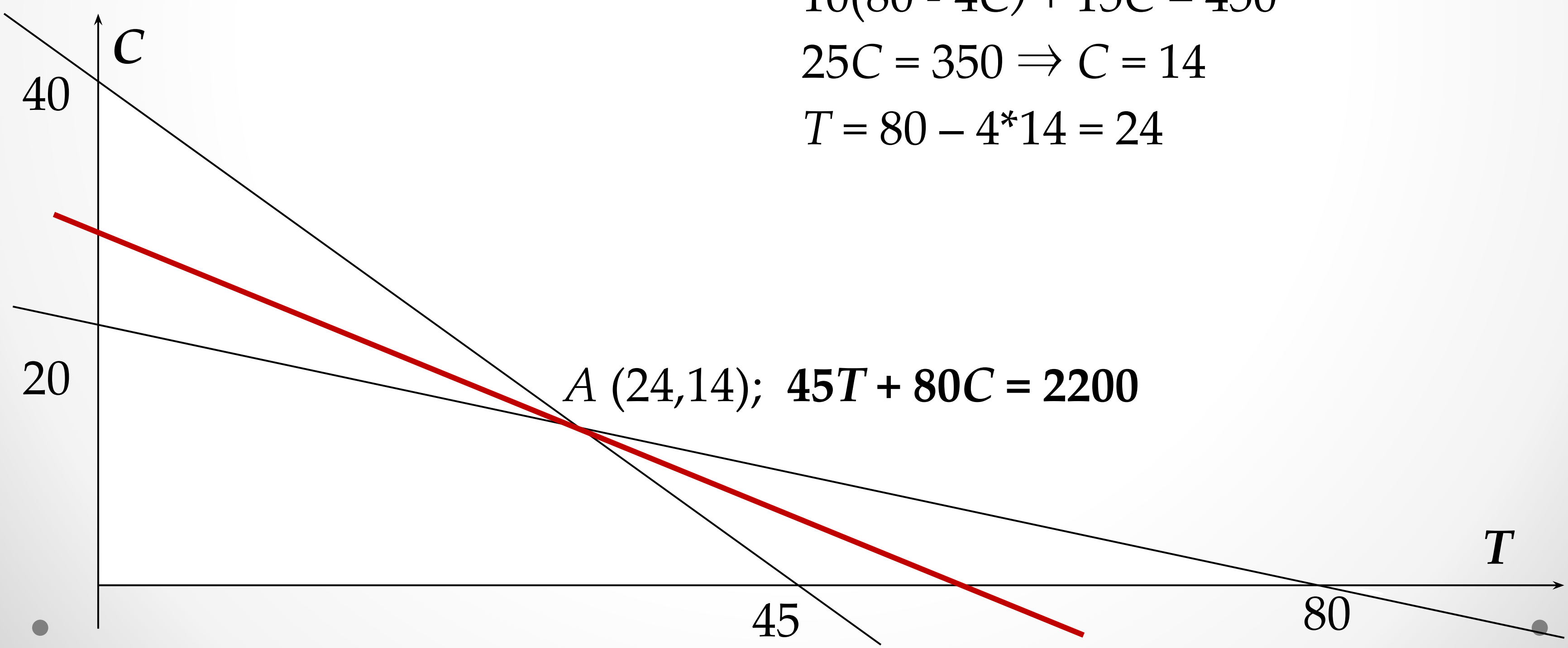
Из (1) $T = 80 - 4C$

Тогда (2) принимает вид:

$10(80 - 4C) + 15C = 450$

$25C = 350 \Rightarrow C = 14$

$T = 80 - 4*14 = 24$



Линейное программирование

(Двойственная задача)

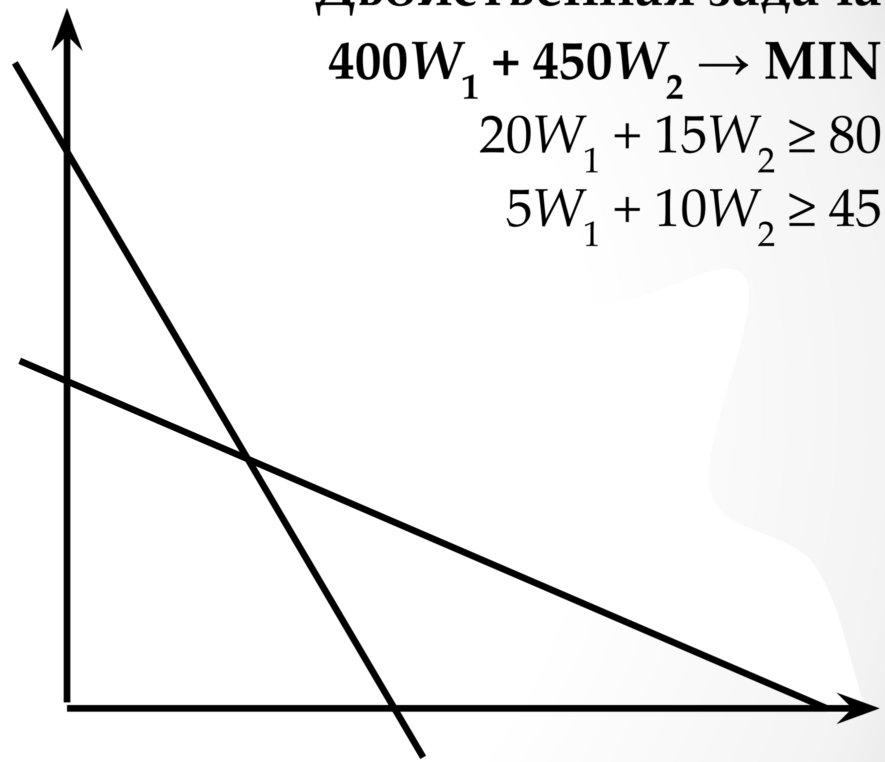
Произ-водст-венная задача	Стол (С)	Табурет (Т)	Ограничение (R)	Двойственная задача
Материал	20	5	400	**W_1**
Работа	15	10	450	**W_2**
Прибыль	80	45	Max	**Огр. Min**

Двойственная задача

$$400W_1 + 450W_2 \rightarrow \mathrm{MIN}$$

$$20W_1 + 15W_2 \geq 80$$

$$5W_1 + 10W_2 \geq 45$$



W_1 и W_2 – стоимость материала и работы (труда), оцененные по вкладу в целевую функцию

Линейное программирование

(Еще одна задача)

Задача о раци- оне	Корм M В 1 кг.	Корм N В 1 кг.	Надо
Вит. А	0,1	0,25	1
Вит. В	1	0,25	5
Калории	110	120	400
Цена 1 кг	3,8	4,2	MIN

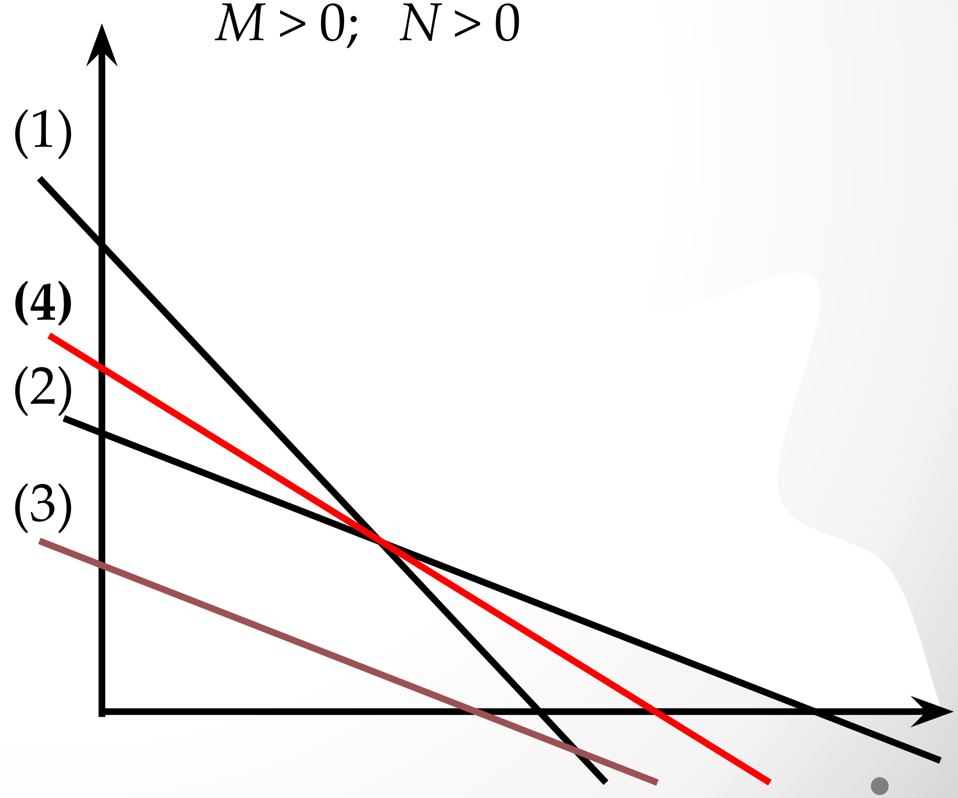
$$\mathbf{3{,}8}\boldsymbol{M} + \mathbf{4{,}2}\boldsymbol{N} \rightarrow \mathbf{MIN} \quad \mathbf{(4)}$$

$$0{,}1M + 0{,}25N \geq 1 \quad (1)$$

$$1M + 0{,}25N \geq 5 \quad (2)$$

$$110M + 120N \geq 400 \quad (3)$$

$$M > 0; \quad N > 0$$



Линейное программирование

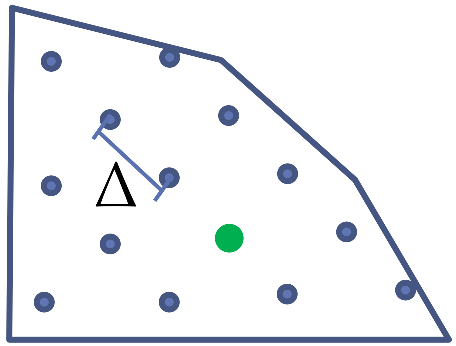
(Методы решения задач – на примере двумерных (с 2 переменными))

Нахождение квадратного корня из 3 методом перебора

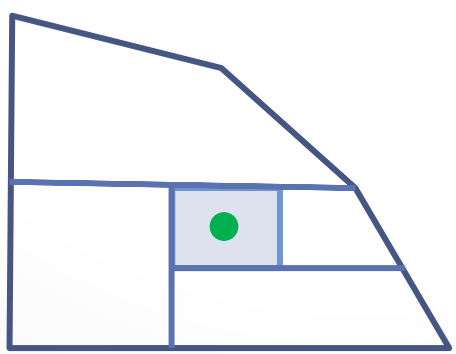
x(i)	F(x(i))
1	2
1,1	1,79
1,2	1,56
1,3	1,31
1,4	1,04
1,5	0,75
1,6	0,44
1,7	**0,11**
1,8	**-0,24**
1,9	-0,61
2	-1

$$f(x) = 3 - x_i^2$$

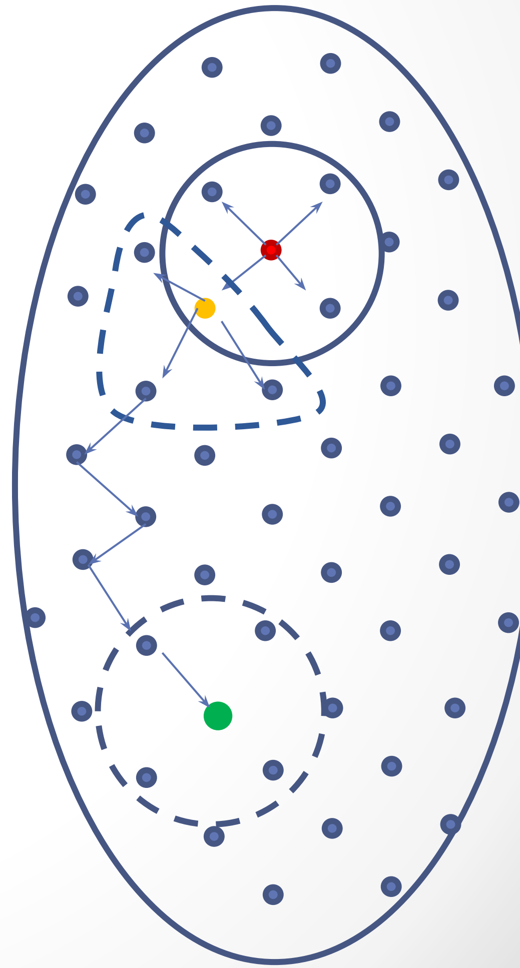
Простой перебор



Направленный перебор



Симплекс-метод



Линейное программирование
(ТРАНСПОРТНАЯ ЗАДАЧА)

	Потреб. 1	Потреб. 2	Потреб. 3	Потреб. 4	Запас на складах
Склад 1	2	5	5	5	60
Склад 2	1	2	1	4	80
Склад 3	3	1	5	2	60
Потребн.	50	40	70	40	200

Целевая функция:

$$F = 2X_{11} + 5X_{12} + 4X_{13} + 5X_{14} + X_{21} + 2X_{22} + X_{23} + 4X_{24} + 3X_{31} + X_{32} + 5X_{33} + 2X_{34} \rightarrow \mathrm{MIN}$$

Ограничения:

$$X_{11} + X_{12} + X_{13} + X_{14} = 60$$

$$X_{21} + X_{22} + X_{23} + X_{24} = 80$$

$$X_{31} + X_{32} + X_{33} + X_{34} = 60$$

$$X_{11} + X_{21} + X_{31} = 50$$

$$X_{12} + X_{22} + X_{32} = 40$$

$$X_{13} + X_{23} + X_{33} = 70$$

$$X_{12} + X_{22} + X_{32} = 40$$

- Существует программная реализация решения подобных задач

Целочисленное программирование

(Дискретная задача. Решения – целые числа)

Выбор обору-дования	Станок А (X)	Станок Б (Y)	Огранич ен. (C)
Площадь м. кв.	8	4	38
Цена тыс. долл.	5	2	20
Производит Тыс. шт.	7	3	Max

ПОСТАНОВКА ЗАДАЧИ

$\mathbf{7X + 3Y \rightarrow MAX}$

$8X + 4Y \le 38$

$X \ge 0; \quad Y \ge 0; \quad X,Y$ - целые

$5X + 2Y \le 20$

Возможно решение

<u>МЕТОДОМ ПЕРЕБОРА</u>

Из ограничений $\Rightarrow \quad X \le 4.$

1. $X = 4 \quad \Rightarrow \quad Y = 0 \quad \Rightarrow \quad C = 28$
   (выбирается min Y, при котором выполняются оба ограничения).
2. $X = 3 \quad \Rightarrow \quad Y = 2 \quad \Rightarrow \quad C = 27.$
3. <u>**$X = 2 \quad \Rightarrow \quad Y \le 5 \quad \Rightarrow \quad C = 29.$**</u>
4. $X = 1 \quad \Rightarrow \quad Y \le 7 \quad \Rightarrow \quad C = 28.$
5. $X = 0 \quad \Rightarrow \quad Y \le 9 \quad \Rightarrow \quad C = 27.$

Для обеспечения MAX производительности покупаем 2 станка типа А и 5 станков типа В

Целочисленное программирование
(ЗАДАЧА О РАНЦЕ)

i	1	2	3	4	R = 6
a_i	3	2	4	5	
c_i	2	1	3	4	

Суть задачи сводится к тому, чтобы поместить в ранец ограниченного объема набор предметов максимальной полезности.

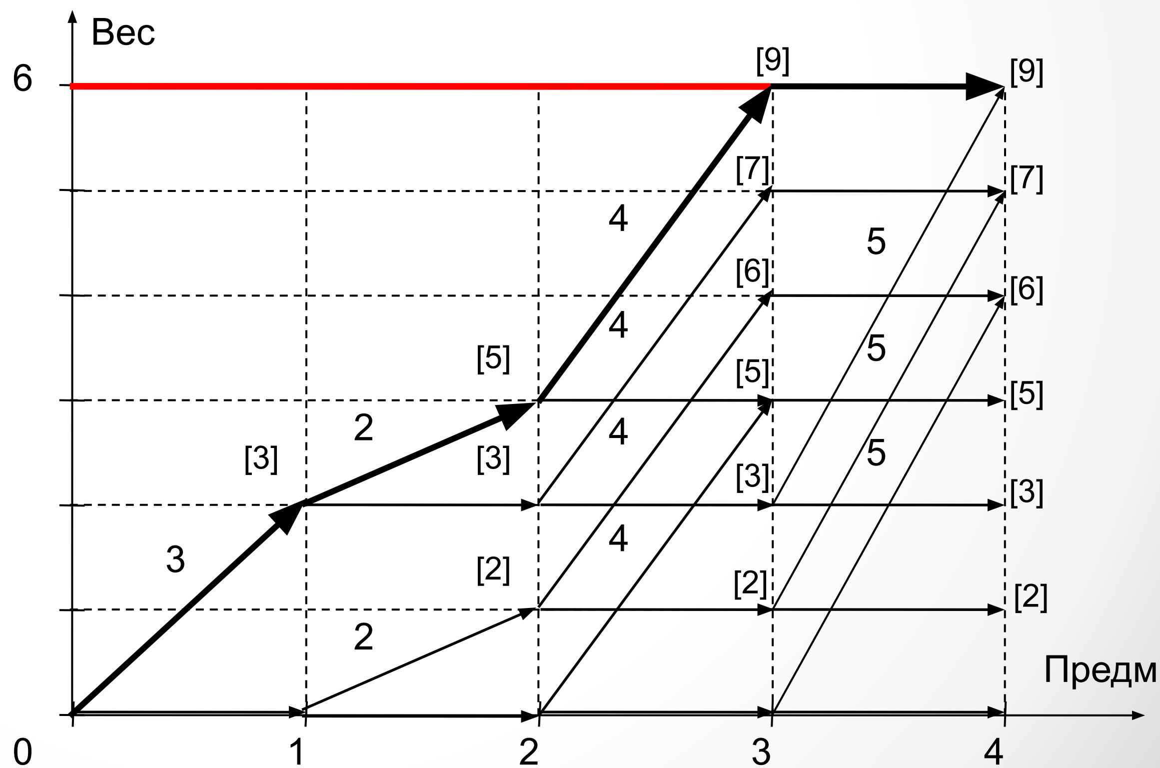
Каждый предмет имеет свой объем (c_i) и полезность (a_i).

Переменные x_i равны 1 если i-й предмет входит в конечный набор, и $x_i = 0$, если не входит.

Приведем математическую постановку задачи:

$$\sum_{i \in Q} \alpha_i x_i \rightarrow \max$$

$$\sum_{i \in q} c_i x_i \leq R$$



Комбинаторное программирование

NP-трудные задачи – не полиномиальные (экспонента – 2^n) –перебором решить невозможно.

Примеры NP-трудных задач:

Задача о ранце. – *последовательность* из n элементов, каждый из которых может принимать 2 значения ($x_i = [0;1]$)) Число комбинаций 2^n

Задача коммивояжера – *перестановка* из n элементов $\pi=(i^1, i^2, \dots, i^n)$ ($n!$).

Подбор команды численностью m человек из n претендентов. (Число *сочетаний*).

Примеры комбинаций

Перестановка (P_n) из n элементов (например чисел 1,2,…,n) – любой упорядоченный набор из элементов. Размещение из n элементов по n – $\boldsymbol{n!}$.

Размещение (A_n^k) из n элементов по k – упорядоченный набор из k различных элементов некоторого n-элементного множества – $\boldsymbol{n!/(n-k)!}$.

Сочетание из n по k (C_n^k) – набор k элементов, выбранных из данных n элементов. Наборы, отличающиеся только порядком следования элементов (но не составом), считаются одинаковыми (отличие от размещений) – $\boldsymbol{n!/(n-k)!/k!}$.

Последовательность – набор из n элементов, каждый из которых может принимать одно из m значений – $\boldsymbol{m^n}$.

$$20! = 2{,}432902 \times 10^{18}$$

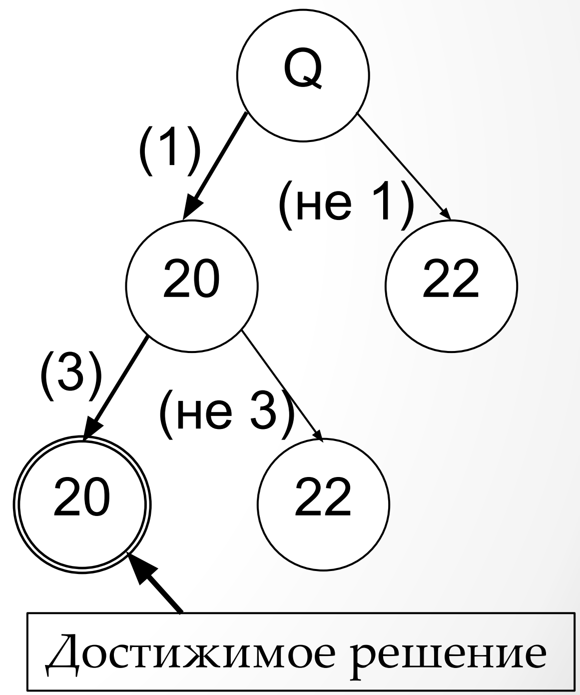
(Проверяя по 100 комбинаций в 1 сек. мы потратим около 770 млн. лет)

Целочисленное программирование
(Некоторые методы решения задач)

Метод приближения непрерывными задачами – сначала решается задача линейного программирования без учета целочисленности, а затем в окрестности оптимального решения ищутся целочисленные точки.

Метод ветвей и границ (один из наиболее известных видов *метода направленного перебора*) – множество возможных решений делится на непересекающиеся подмножества, для каждого вычисляется верхняя (задача на MAX) или нижняя (задача на MIN) граница. Выбирается подмножество с максимальной верхней или минимальной нижней границей и процедура повторяется для него, пока не получим оптимальное решение или решение с необходимой точностью.

Эффективность метода ветвей и границ в существенной степени зависит от «качества» оценок. При плохих оценках это фактически полный перебор, при достижимой нижней оценке – это получение оптимального решения за один проход по дереву ветвлений.

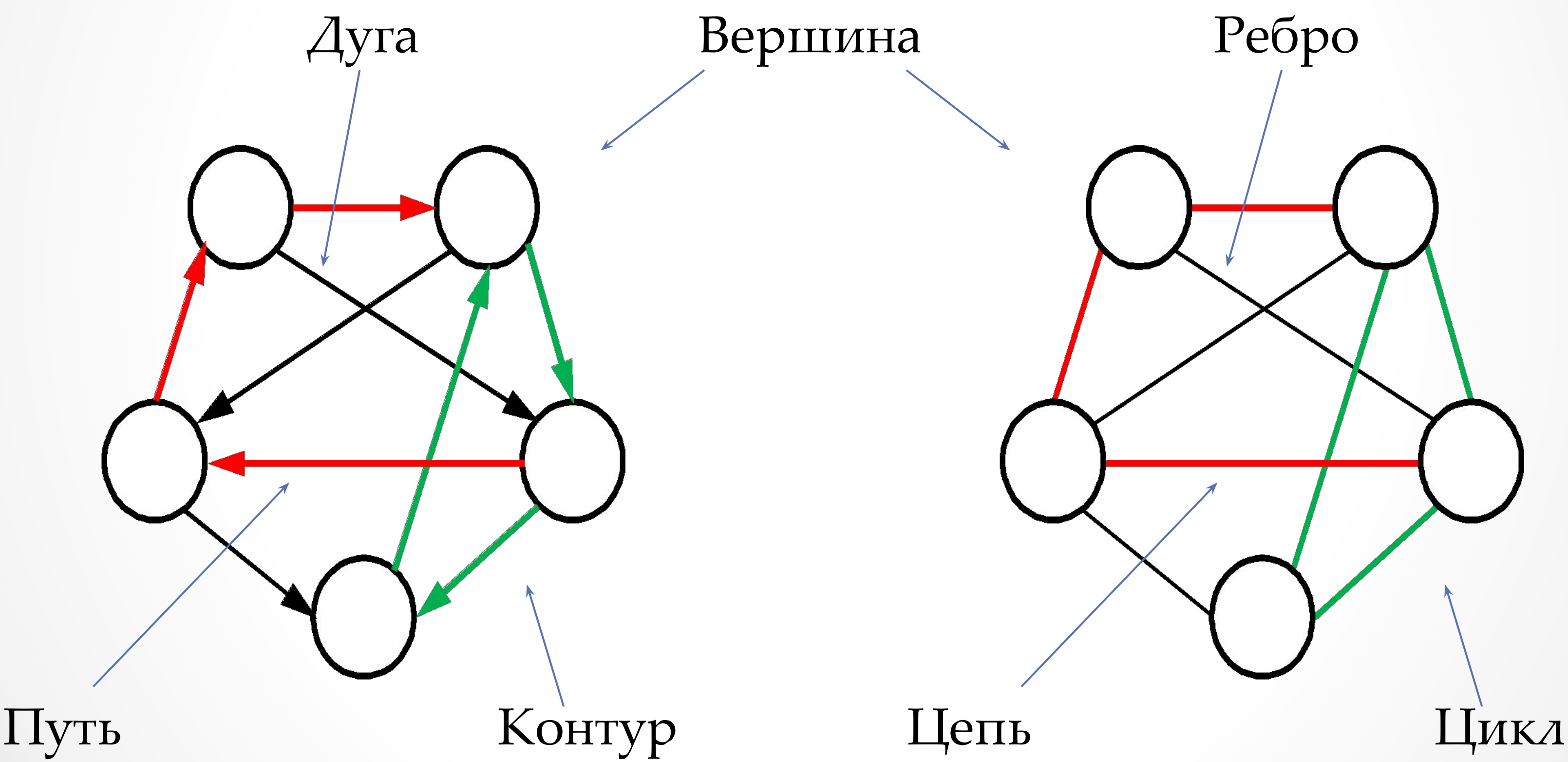


Задача об очередности обработки 3 деталей на станках

Теория графов и оптимизация
(Основные понятия)

ОРИЕНТИРОВАННЫЙ ГРАФ

НЕОРИЕНТИРОВАННЫЙ ГРАФ



Теория графов и оптимизация
(ЗАДАЧА О КРАТЧАЙШЕМ (наидлиннейшем) ПУТИ)

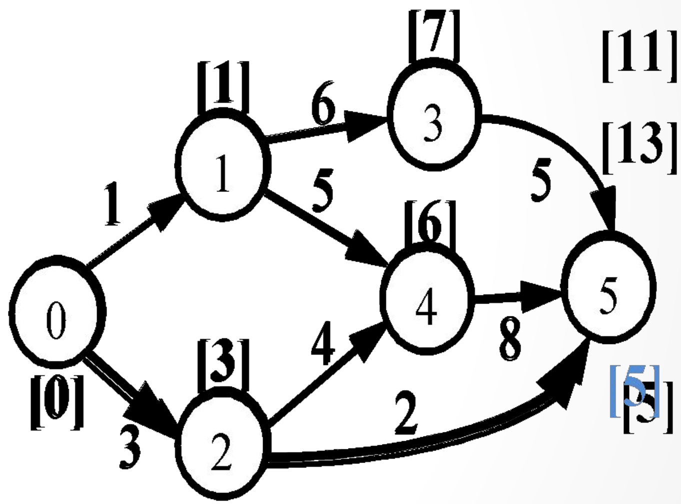
НАЙТИ КРАТЧАЙШИЙ ПУТЬ В ГРАФЕ

На ребрах указаны их длины.

В квадратных скобках рядом с вершинами – MIN длина пути в эту вершину.

Путь MIN длины ищется методом обратного хода.

У задачи о кратчайшем пути есть массу интерпретаций:
- календарное планир.,
- последовательность работ,
- собственно поиск кратчайшего пути
- и многие другие.



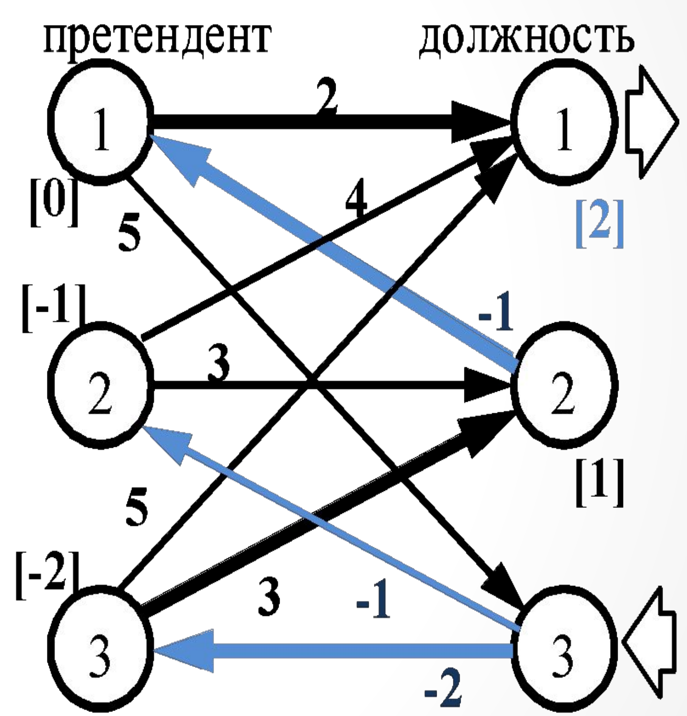
Теория графов и оптимизация
(ЗАДАЧА О НАЗНАЧЕНИИ)

НАЙТИ ОПТИМАЛЬНОЕ НАЗНАЧЕНИЕ НА ДОЛЖНОСТИ

Должн. / Прет.	1	2	3
1	2	<u>1</u>	5
2	4	3	<u>1</u>
3	5	3	<u>2</u>

В таблице – «зарплата», которую запрашивает претендент.

Эта задача может решаться с помощью алгоритма поиска кратчайшего пути, рассмотренного выше.



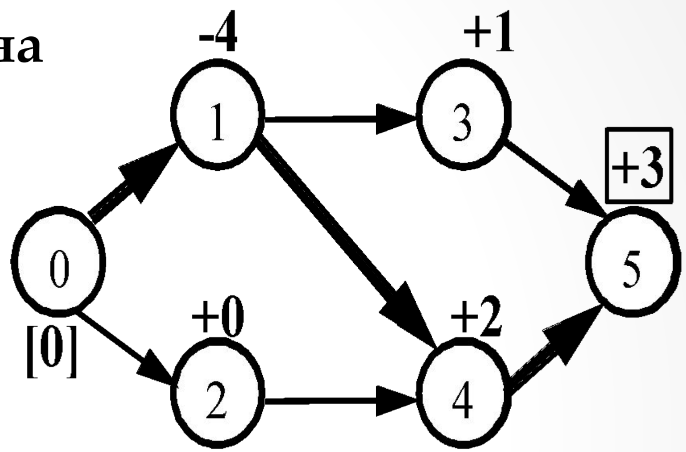
Двудольный граф

РЕШЕНИЕ: П1 – Д1 Min затр. – 4
П2 – Д3 Возр. на 2
П3 – Д2 Стало – 6

Теория графов и оптимизация
(ЗАДАЧА О МАКСИМАЛЬНОМ ПОТОКЕ)

НАЙТИ ПОТОК MAX ВЕЛИЧИНЫ

Алгоритм (теорема) Форда-Фалкерсона

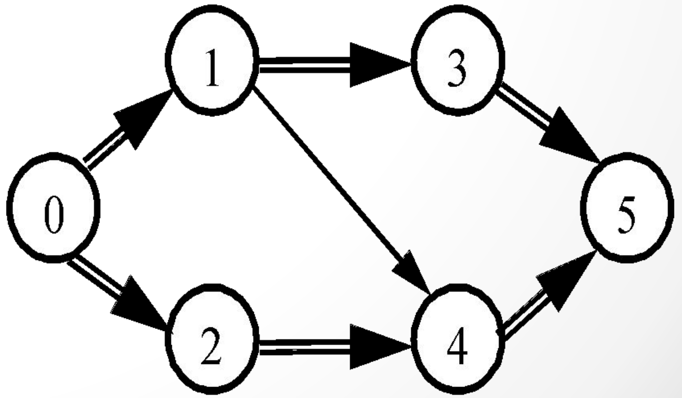


Пусть пропускные способности всех дуг равны 1.

Пропускаем произвольный поток.

Помечаем вершины, начиная с входа «0»

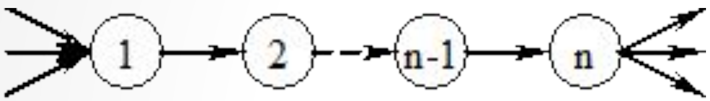
- если можно увеличить поток в к.-л. вершину, помечаем ее (поток увеличивается на 1).
- Если в помеченную вершину входит поток из непомеченной, его можно вычесть (уменьшить).



Если в конце помечена вершина-выход, поток м.б. увеличен, если нет – поток максимален.

(ЗАДАЧА О МАКСИМАЛЬНОМ ПОТОКЕ. ОПРЕДЕЛЕНИЯ)

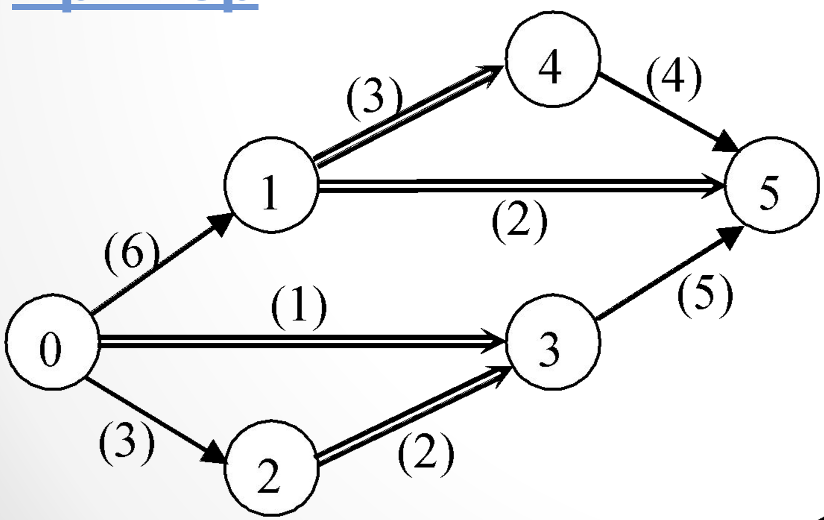
Последовательным множеством называется подмножество дуг графа, образующих путь такой, что любая вершина, кроме начальной, имеет степень захода 1, и любая вершина, кроме конечной имеет степень исхода 1.

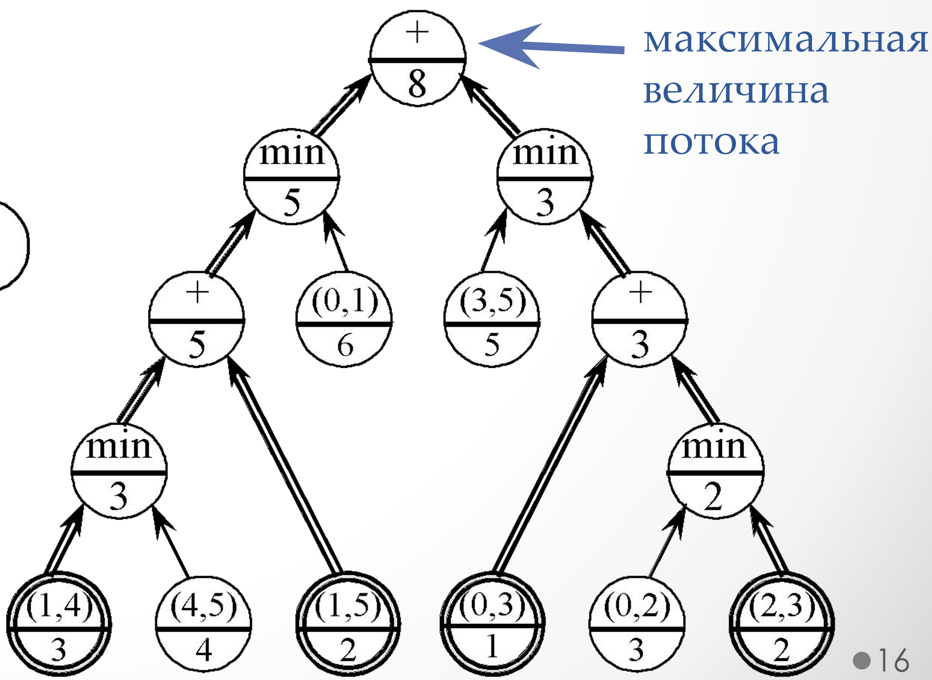


Параллельным множеством дуг называется подмножество дуг сети, у которых начальная и конечная вершины совпадают.

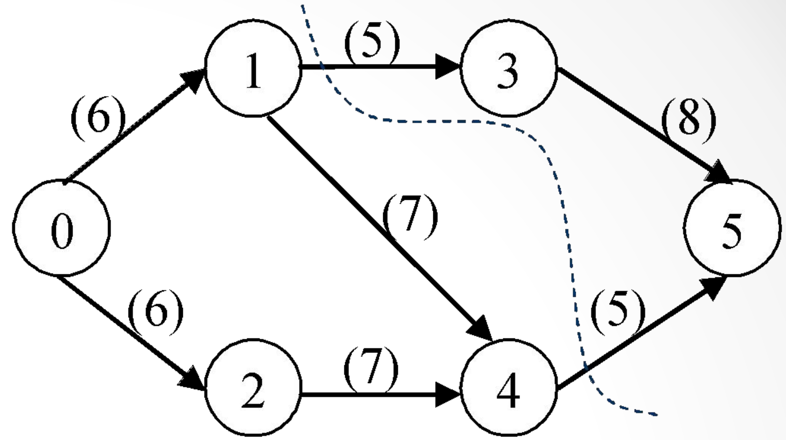
Сеть агрегируемая, если путем агрегирования последовательных и (или) параллельных множеств дуг ее можно свести к одной дуге.

Пример



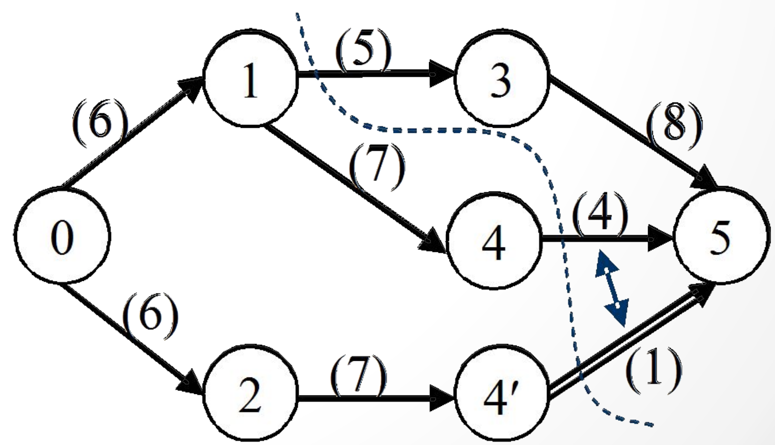


Теорема 1. Величина максимального потока в агрегируемой сети меньше или равна величине максимального потока в исходной сети. Доказательство следует из очевидного факта, что любому потоку в агрегируемой сети однозначно соответствует некоторый поток в исходной сети.



MAX поток = MIN разрез

Теорема 2 (двойственности). Существует разбиение пропускных способностей дуг, исходящих из разделенных вершин, такое что величина максимального потока в агрегируемой сети равна величине максимального потока в исходной сети **(ОДЗ)**

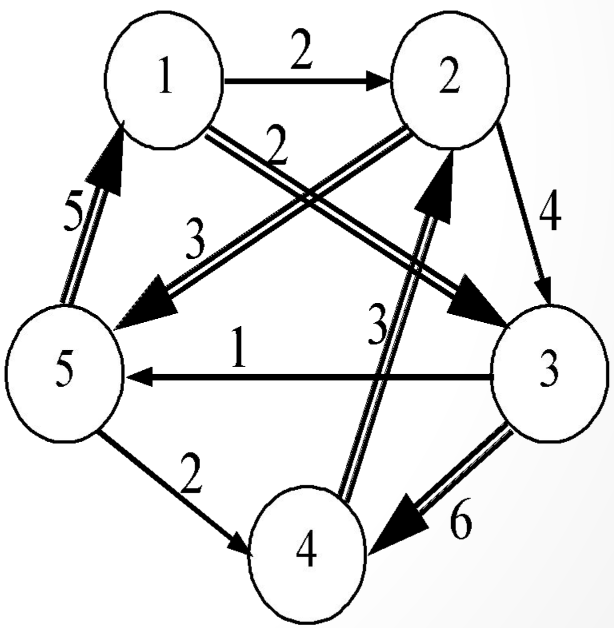


Теория графов и оптимизация
(ЗАДАЧА КОММИВОЯЖЕРА)

Найти кратчайший путь, соединяющий все города с возвратом в исходный
или
Найти КОНТУР МИНИМАЛЬНОЙ ДЛИНЫ

Задача – NP-трудная. Для ее решения могут применяются:

- Эвристические методы (напр. Всегда идти в ближайший из не пройденных городов)
- Метод локальной оптимизации
- Метод ветвей и границ



Критерии принятия решений
(МАТЕМАТИЧЕСКИЕ ПОСТАНОВКИ)

Сценарии				
		B_2	… B_j …	Bm
A_1		U_{12}	…	U_{1m}
… A_i …		…	U_{ij} (полезность)	…
A_N		U_{N2}	…	U_{Nm}

1. *Максиминный критерий Вальда* (наибольшей осторожности)

$$U^* = \max_i \min_j U_{ij}$$

2. *Критерий минимаксного сожаления* (критерий Сэвиджа).

$$C_{ij} = \max_i U_{ij} - U_{ij} \quad \text{- сожаление}$$

$$U^* = \min_i \max_j C_{ij} \quad \text{- выбранная альтернатива}$$

3. *Критерий максимакса* (крайний оптимизм):

$$U^* = \max_i \max_j U_{ij}$$

4. *Критерий Гурвица*. Пусть для *i*-ой альтернативы $m_i = \min_j U_{ij}; M_i = \max_j U_{ij}$

$$U_i(\alpha) = \alpha m_i + (1-\alpha) M_i, \quad 0 \le \alpha \le 1$$

При заданном α (степень оптимизма ЛПР) выбирается $U^* = \max_i U_i(\alpha)$.
При $\alpha = 1$ данный критерий переходит в *критерий Вальда*.

5. *Критерий Лапласа*. (Аналог вероятностной модели при равных вероятностях.)
Для каждой альтернативы *Ai* определяется показатель

$$U_{icp} = (1/m)\sum_{j=1}^{m} U_{ij}$$

Далее выбирается $U^* = \max_i U_{icp}$

.