Лекция 5. Задачи мультипроектного управления

Задача о ранце, как формализация задачи выбора оптимального портфеля проектов. Одномерная и многомерная задачи о ранце. Задача выбора варианта реализации программы и ее решение методом дихотомического программирования. Сетевые графики с рекомендательными зависимостями работ. Задача оптимизации последовательности выполнения проектов

Предисловие

- Три уровня задач управления проектами
  - Задачи мультипроектного управления (проект как единое целое)
  - Задачи календарно-сетевого планирования, то есть планирования последовательности выполнения работ в рамках одного проекта (работа рассматривается как единое целое)
  - Задачи управления интенсивностью работ (можно менять параметры отдельной работы)

Задачи мультипроектного управления

- Типичный случай – выбор портфеля независимых проектов при общем ресурсном ограничении = задача о ранце
  - Имеется множество предметов $N = \{1, \dots, n\}$ с весами $c_i, i = 1, \dots, n$ и ценностями p_i, $i = 1, \dots, n$, и рюкзак емкости *R*.
  - Необходимо выбрать подмножество предметов *N*' суммарным весом не более *R*, имеющее максимальную ценность.
  - $\sum_{i=1}^{n} x_i \cdot p_i \to max$ при $\sum_{i=1}^{n} x_i \cdot c_i \leq R$,

    $x_i \in \{0,1\}$
- В реальности возникают более сложные задачи => требуются обобщения задачи о ранце

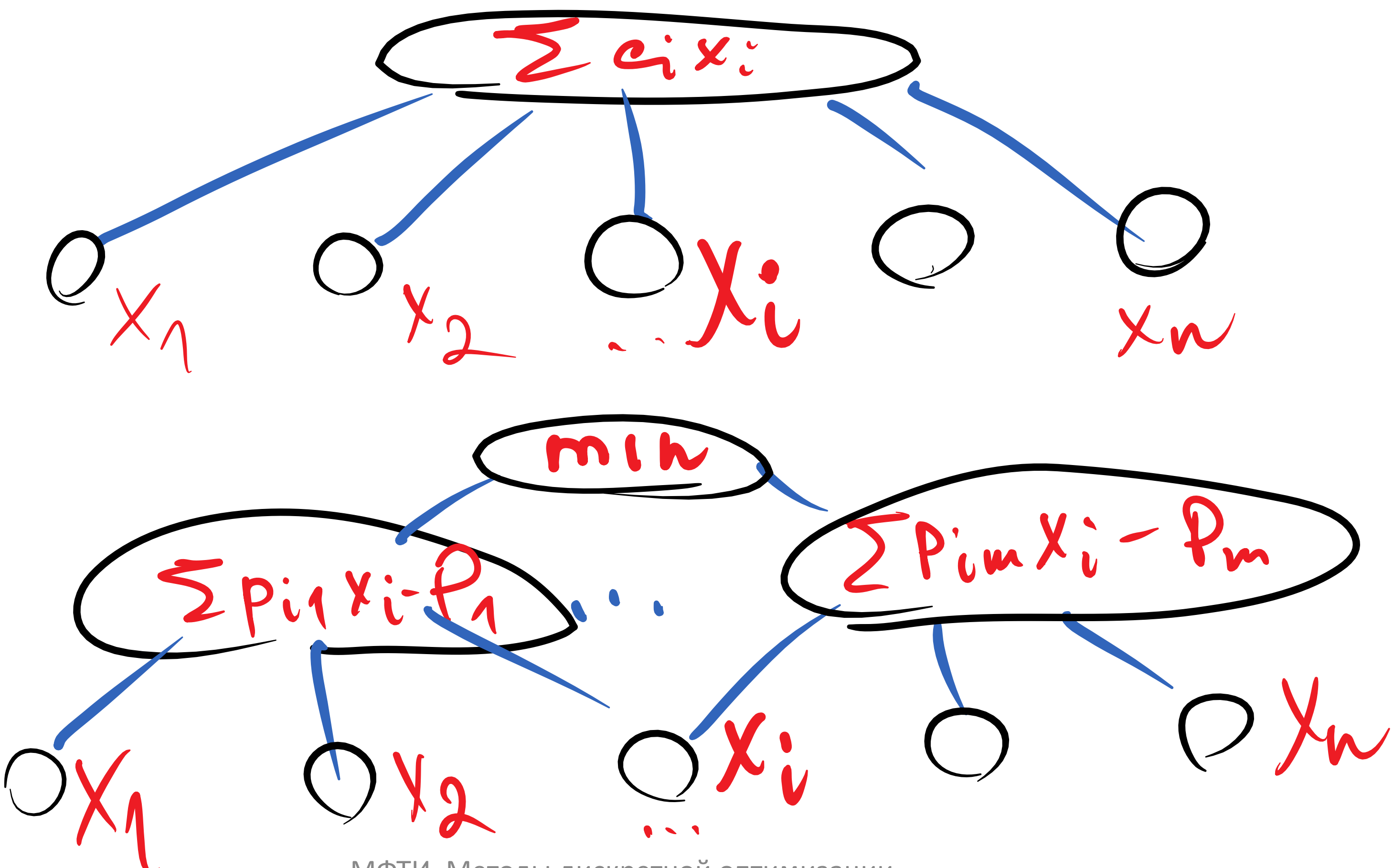
Задачи мультипроектного управления

- Например, задача минимизации программы по стоимости
  – Имеется множество проектов $N = \{1, \dots, n\}$ с затратами c_i, $i = 1, \dots, n$, и величинами эффекта p_{ij}, $i = 1, \dots, n$, $j = 1, \dots, m$, по m критериям («социалка», экономика, экология и т.п.).
  – Необходимо отобрать набор проектов минимальной стоимости $\sum_{i=1}^{n} x_i \cdot c_i \to min$ так, чтобы по каждому критерию $j = 1, \dots, m$ достигался заданный эффект P_j: $\sum_{i=1}^{n} x_i \cdot p_{ij} \geq P_j$, $j = 1, \dots, m$,

    $x_i \in \{0,1\}$.
- В случае одного критерия – это обратная задача к задаче о ранце
- В случае нескольких критериев задача решается методом сетевого программирования

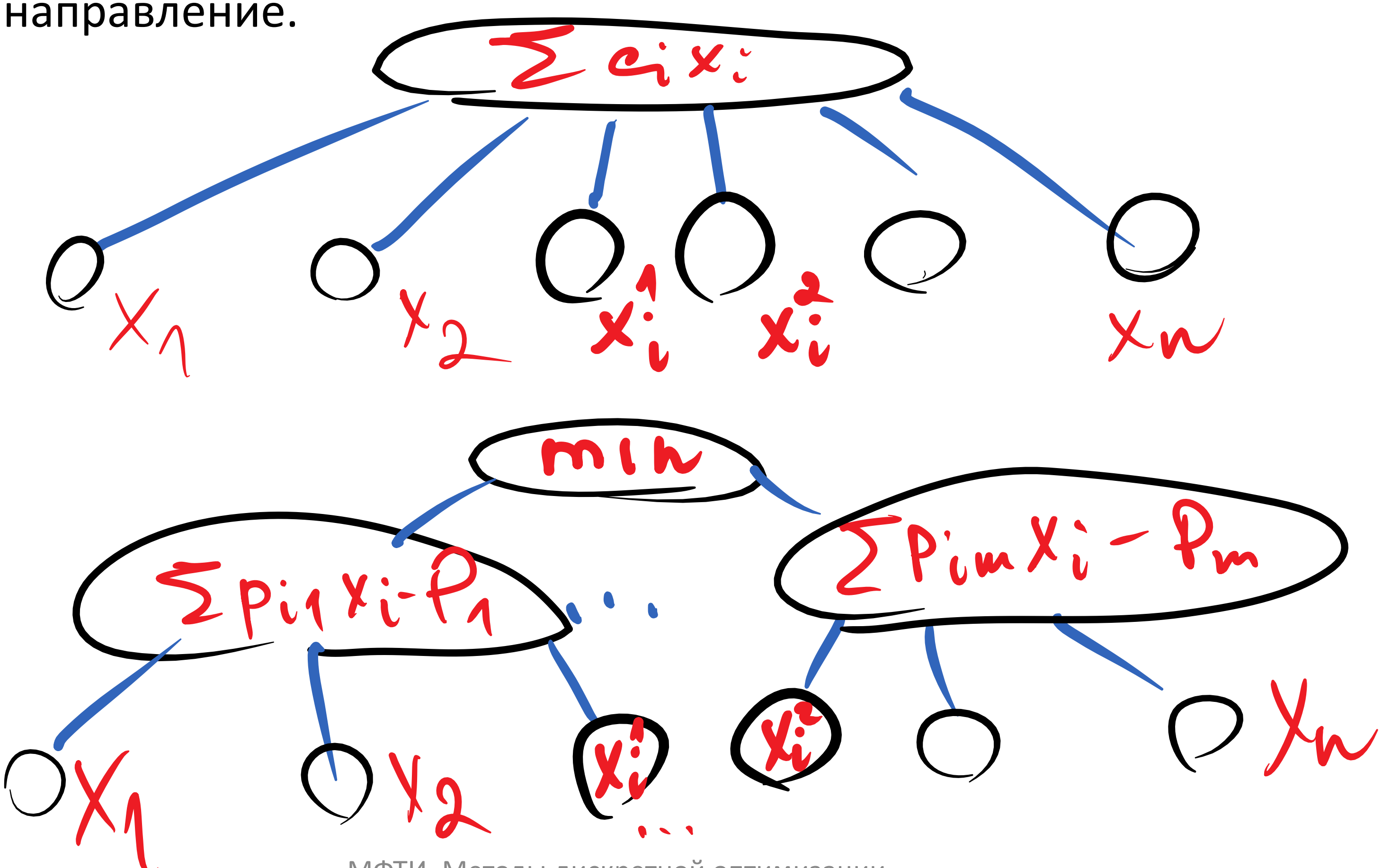
Задача минимизации программы по стоимости

- Сетевое представление критерия и ограничений задачи



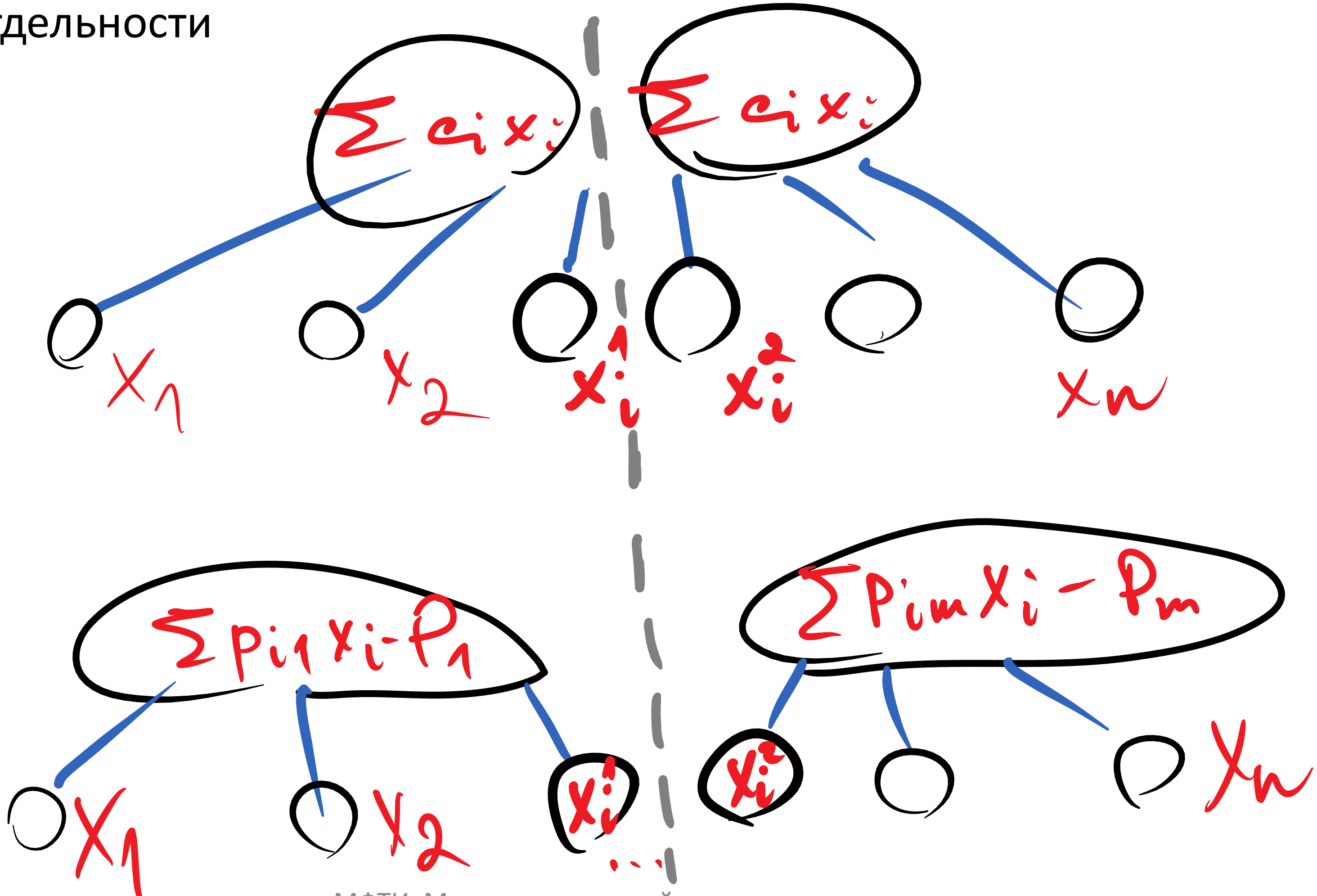
Задача минимизации программы по стоимости

- Каждый проект *i*, дающий вклад в несколько направлений, разобьем на несколько проектов так, чтобы $\sum x_j{}' c_j' = c_i$, и каждый давал вклад в одно направление.



Задача минимизации программы по стоимости

- Можно разбить на независимые задачи о ранце, которые решаются по-отдельности



Задача минимизации программы по стоимости

- Сумма оптимальных значений критерия для этих задач

$$\sum_j \sum_i c_i^j \tilde{x}_i^j$$

или их нижних оценок (например, по методу затраты-эффект)

$$\sum_j \sum_i c_i^j \bar{x}_i^j$$

дает нижнюю оценку для исходной задачи

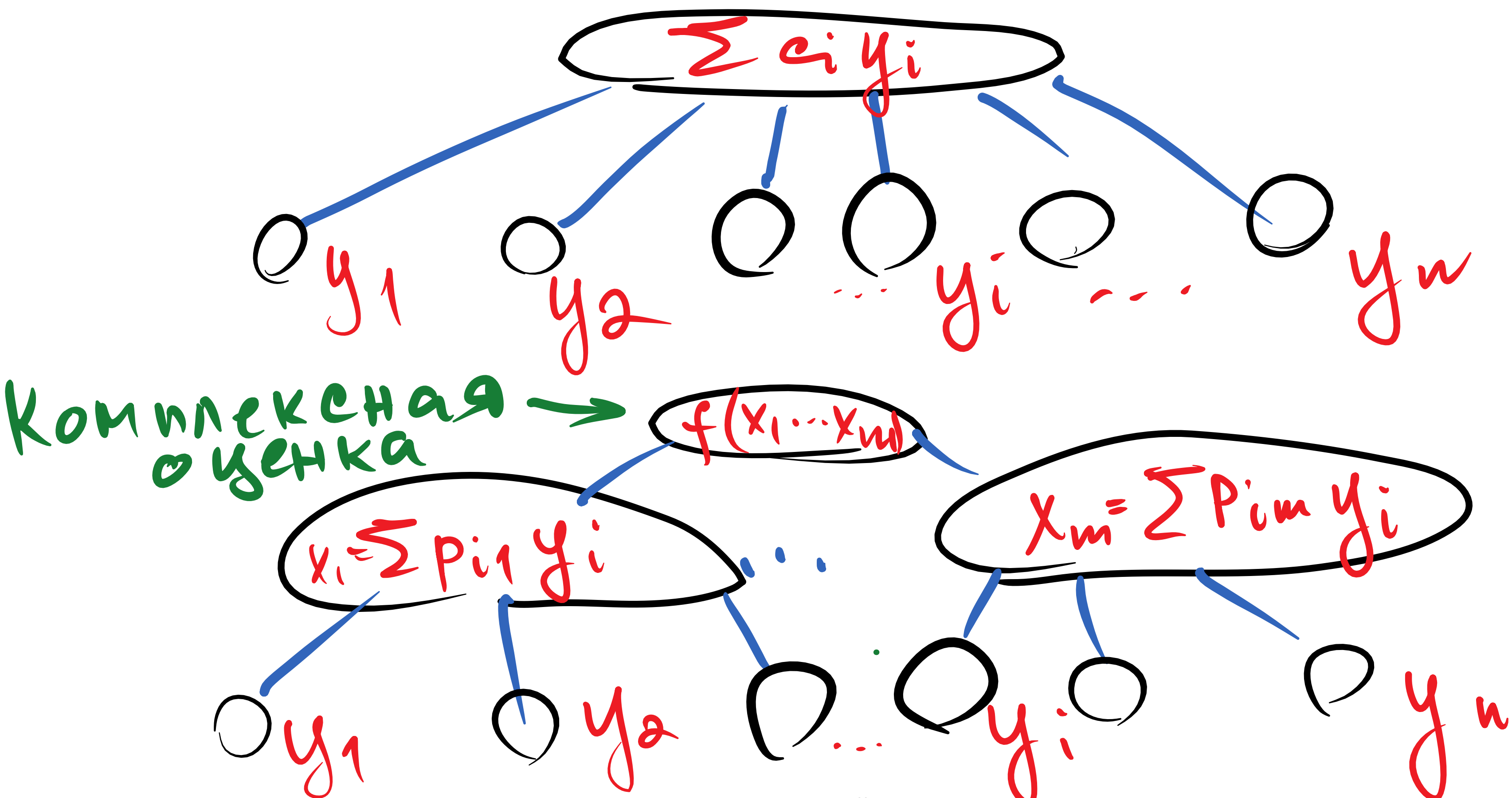
- Ставится двойственная задача – задача максимизации нижней оценки выбором коэффициентов c_i^j разделения проектов
- Это задача выпуклого программирования. Как ее решать?
  - На примере проекта, вносящего вклад в два направления:
    - Если $\tilde{x}_i^1 = \tilde{x}_i^2$, можно ли что-то оптимизировать?
    - Если $\tilde{x}_i^1 = 1, \tilde{x}_i^2$=0, то в каком направлении надо изменить c_i^1 чтобы увеличить оценку?

Задачи мультипроектного управления

- Похожая задача – выбор оптимальной программы развития
  - Задан набор направлений развития. Для каждого направления задан набор мероприятий (проектов развития), с известными затратами и эффектом для этого направления. Необходимо максимизировать комплексный критерий развития – древовидную биматричную свертку критериев различных направлений при ограничении на общий бюджет программы (сумму затрат выбранных мероприятий-проектов).
  - В случае одного направления – задача о ранце с дискретной шкалой ценности
  - Для нескольких направлений решается методом дихотомического программирования
  - Для многоцелевых проектов – сетевое программирование

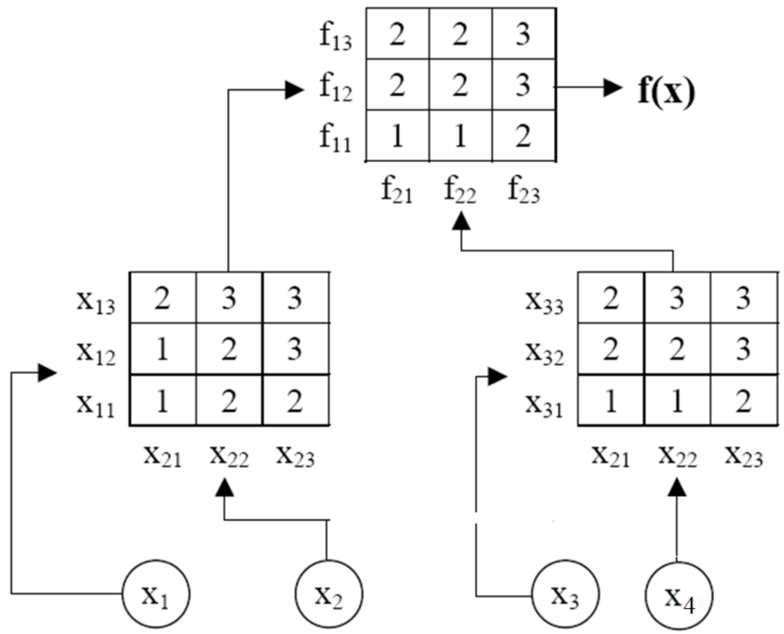
Выбор оптимальной программы развития

- Сетевое представление этой задачи очень похоже на сетевое представление предыдущей задачи.



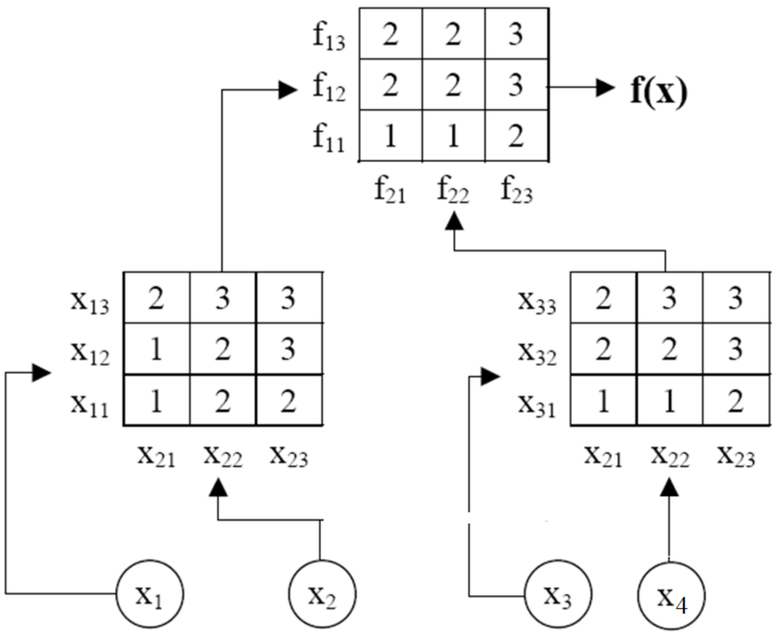
Выбор оптимальной программы развития

- Комплексная оценка задается биматричной сверткой, допускающей дихотомическое представление
- Сворачиваются оценки, принимающие одно из *k* значений
- Матрицы монотонны (увеличение каждого из частных критериев не уменьшает значения свертки)
- Предыдущая задача – частный случай для свертки типа «минимум»
- Если каждый проект вносит вклад только в одно направление, получили задачу дихотомического программирования



Выбор оптимальной программы развития

- Для каждого из *m* направлений решаем *k* задач минимизации затрат на достижение каждого из *k* значений критерия по *j*-му направлению
- Поднимаясь по дереву, определяем минимальные затраты на достижение каждого из значений каждого из промежуточных критериев, и, наконец, комплексного критерия.
- Обратным ходом восстанавливаем решение

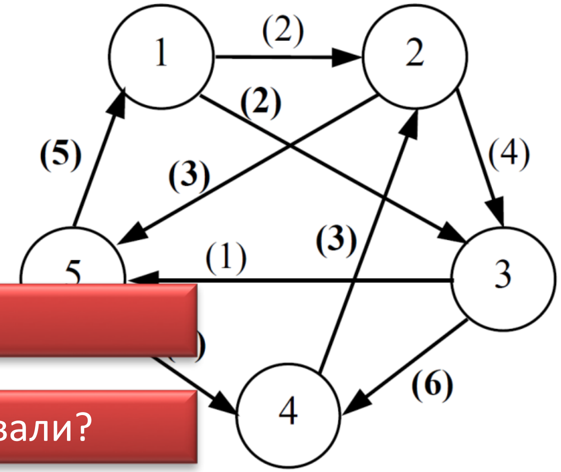


Если некоторые проекты вносят вклад в несколько проектов, получаем задачу сетевого программирования

Другие задачи мультипроектного управления

- Затем появляется время – некоторые проекты длиннее, некоторые – короче. Это решается приведением затрат и эффекта к одному моменту времени (дисконтированию).
- Задача усложняется, когда заданы лимиты финансирования для каждого периода.
- Появляется необходимость финансирования более поздних проектов за счет эффекта завершенных раньше.
- Более простой вариант – результаты одних проектов облегчают последующие (например, инфраструктурные проекты).
- В качестве примера мы рассмотрим одну из самых простых **динамических** задач мультипроектного управления – задачу оптимизации последовательности выполнения проектов

Задача оптимизации последовательности выполнения проектов



- Зададим ориентированный граф $G =< N, E >$, где N – множество проектов, а дуга $ij \in E$ входит в граф если проект *i* рекомендуется завершить раньше проекта *j*. Вес c_{ij} показывает потери при нарушении рекомендо
  проектов.

Сколько всего есть перестановок *n* проектов?

Какие задачи на перестановку мы уже рассматривали?

- Если *G* не и
  реализаци
  рекомендации.

Какая метаэвристика подходит для задач перестановки?

Действительно, в ациклическом графе всегда существует правильная нумерация вершин сети (номер начальной вершины дуги меньше номера конечной). Эта нумерация и определяет оптимальную очередность проектов.

- В общем случае нужно из G удалить некоторое множество V дуг, так что полученный частичный граф не будет иметь контуров и сумма $C(V) = \sum_{ij \in V} c_{ij}$ пропускных способностей удаленных дуг будет минимальной.
- Это задача поиска перестановки проектов

Задача оптимизации последовательности выполнения проектов известна как:

- Задача о наборе обратных дуг минимального веса (минимальной стоимости) – minimum weight (cost) feedback arc set problem
- Задача об ациклическом подграфе максимального веса (максимальной стоимости) – maximum weight (cost) acyclic subgraph problem
- Задача о линейном упорядочении – linear ordering problem (LOP)
  - Найти перестановку проектов чтобы минимизировать вес дуг из поздних проектов в ранние
- Задача о триангуляции (матрицы) – triangulation problem
- Частный случай – задача о медиане Кемени – median order problem
  - Найти упорядочение проектов, максимально учитывающее мнения *m* экспертов. Мнение эксперта – это упорядочение. Наложение *m* упорядочений порождает взвешенный граф. Медиана Кемени дает минимум суммарной ошибки
- Частный случай – задача Слейтера назначения победителей турнира – Slater’s tournament problem

Почему эти задачи эквивалентны?

Задача оптимизации последовательности выполнения проектов

- Задача о максимальном ациклическом подграфе NP-полна даже для равных между собой весов дуг, к которой сводится задача 3-реализуемости.
- Существует гарантированная оценка – любое упорядочение вершин порождает ациклический подграф, вес дуг в котором максимум в два раза меньше, чем в максимальном ациклическом подграфе (какова гарантированная эффективность соответствующего алгоритма?).

Действительно, при любом упорядочении вершин множество дуг делится на две части – правильно ориентированных дуг, и неправильно ориентированных дуг. Как минимум одно из них имеет вес не меньше половины общего веса графа.

- Частный случай: точный алгоритм сложности n^3 для **планарного** графа [Gabow H.N. A presentation for crossing set families…, 1993].
- Для решения этой задачи в общем случае используется алгоритм ветвей и границ. Для него нужны оценки.
- Оценки для задачи о максимальном ациклическом подграфе обычно основаны на релаксации задачи в линейной или квадратичной постановке

Оптимизация последовательности выполнения проектов с помощью сетевого программирования

- Получим нижнюю оценку задачи о минимальном штрафе за нарушение рекомендованной последовательности проектов с помощью метода сетевого программирования.
- 1. Найдем множество U всех элементарных (без самопересечений) контуров графа G.

  Алгоритм Тьернана (самый простой)

  a) Для каждой связной компоненты графа построим дерево всех элементарных (без самопересечений) путей из вершины 1. Все пути, заканчивающиеся в вершине 1, соответствуют элементарным контурам.
  b) Удалим вершину 1 и инцидентные дуги.
  c) Повторим шаг (a) для оставшегося графа, получив все контуры, проходящие через вершину 2, но не проходящие через вершину 1.
- 2. Определим двудольный (что это такое?) граф $H(X, U, E')$,
  - доля вершин которого X соответствуют дугам исходного графа G, доля вершин U – элементарным контурам графа G, а дуги соединяют вершину $ij \in X$ с вершиной $k \in U$, если дуга ij принадлежит k-му контуру.
  - Каждой вершине ij из X сопоставлен вес c_{ij} (вес соответствующей дуги из графа G).
- 3. Задачу о максимальном ациклическом подграфе свели к **задаче о покрытии множествами**:
  - Найти набор вершин $V \subseteq X$ таких, что любая вершина $k \in U$ смежна в H хотя бы с одной $ij \in V$, и $\sum_{ij \in V} c_{ij}$ минимальна.
  - Тоже NP-полная задача.

Решение задачи оптимизации последовательности выполнения проектов

- 4. Поставим задачу как задачу линейного **дискретного** программирования
  - Введем переменные $x_{ij} = 1$ если $ij \in V, x_{ij} = 0$ в противном случае.
  - Введем множества $R_k = \{ij \in X: (ij, k) \in E'\}$ дуг из X, составляющих цикл $k \in U$ в *H*.
  - Задача состоит в том, чтобы найти $\min \sum_{ij \in X} c_{ij} x_{ij}$ при условиях $\sum_{ij \in R_k} x_{ij} \geq 1, k \in U$. (запишите в матричной форме!)
  - Заметим, что число ограничений задачи (равное числу контуров) может быть экспоненциальным по *n*.
- 5. Но метод генерации столбцов позволяет решать непрерывные задачи ЛП с огромным числом переменных (обычно далеко не все переменные значимы, и используются только нужные)
- 6. Для использования МГС получим непрерывную релаксацию **двойственной** задачи

Релаксация задачи оптимизации последовательности выполнения проектов

- Можно брать линейную релаксацию напрямую, но мы получим релаксацию на основе **двойственной задачи линейного программирования** методом сетевого программирования.

Двойственная задача ЛП Для задачи $c^T x \to \min$ при $Ax \le b, x_i \ge 0, i = 1, \ldots, n$ двойственная задача

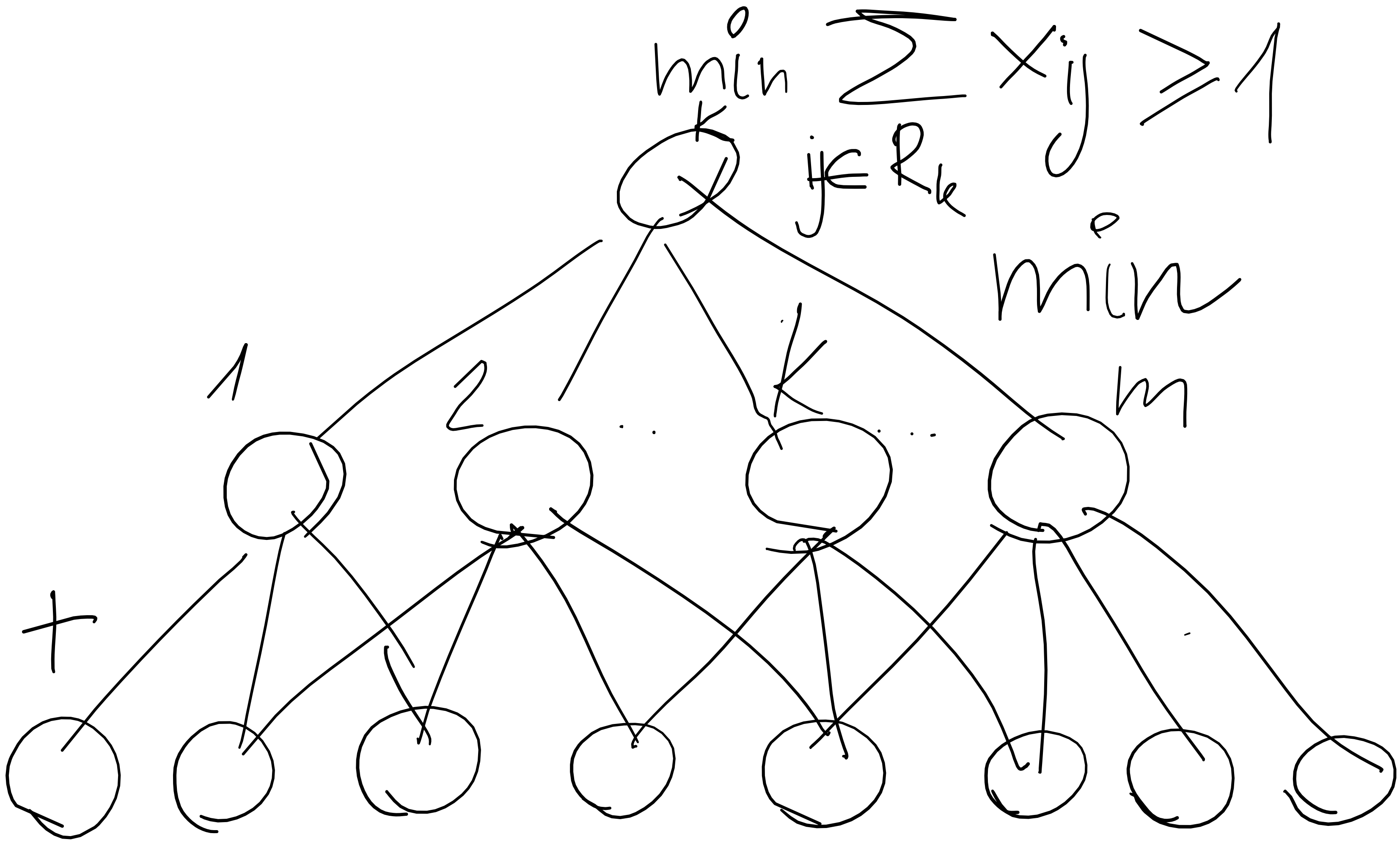
$$b^T y \to \max \quad \text{при } A^T y \ge c, y_i \ge 0, i = 1 \ldots m$$

- Введем веса дуг $s_{ij,k}$ таким образом, чтобы $\sum_{k:(ij,k)\in E'} s_{ij,k} = c_{ij}$.
- Фактически почти каждую переменную x_{ij} разбили на несколько.
- Теперь для каждого k = 1, ..., m задача $\min \sum_{ij\in X} s_{ij,k} x_{ij}$ при условии $\sum_{ij\in R_k} x_{ij} \ge 1$ имеет очевидное решение (какое?)
- Это нижняя оценка для исходной задачи, поскольку значения разделенных переменных в разных частных задачах могут отличаться.

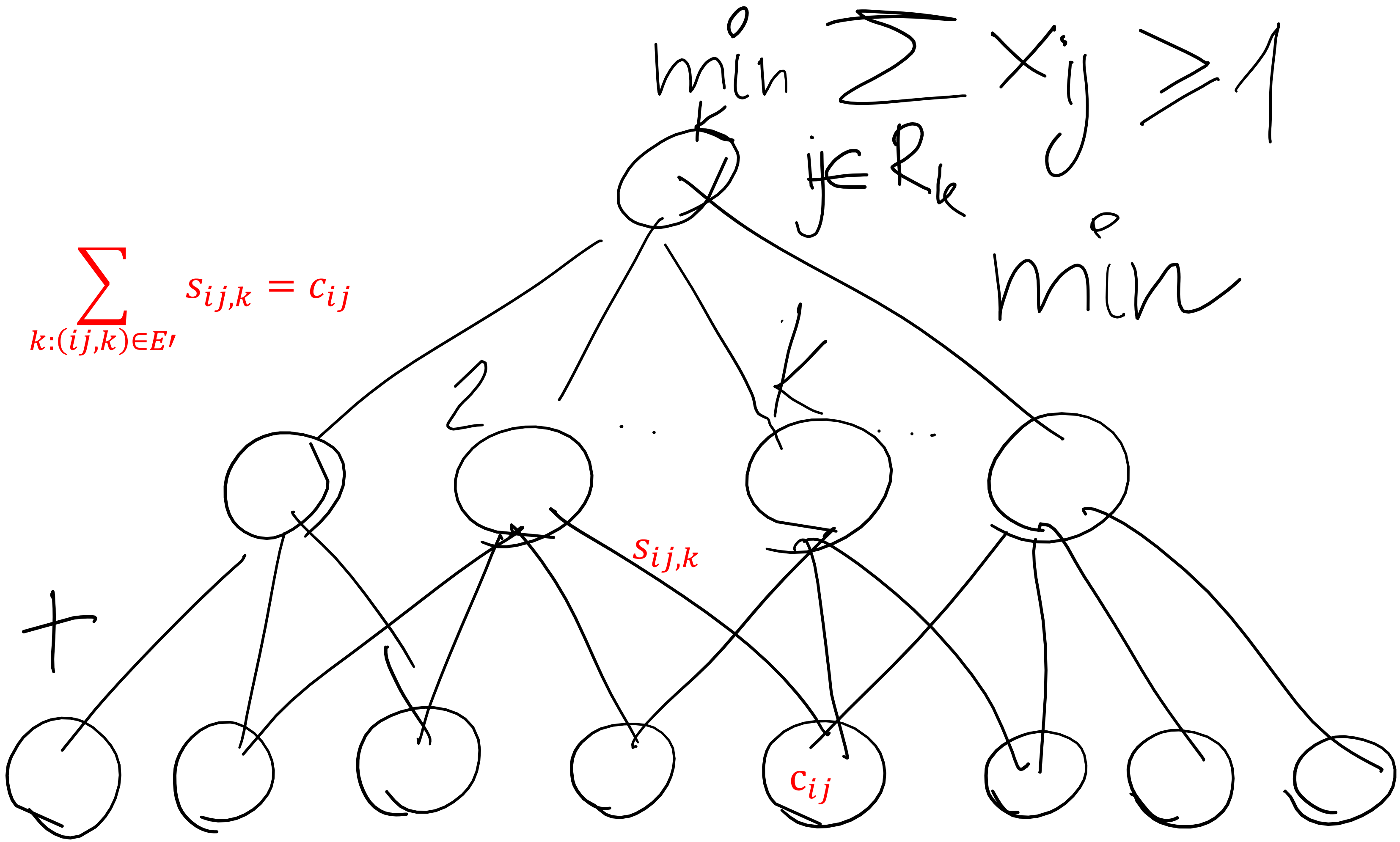
Релаксация задачи оптимизации последовательности выполнения проектов

- Можно поставить двойственную задачу – задачу максимизации этой оценки: $\sum_{k=1}^{m} \min_{ij \in R_k} s_{ij,k} \to \max$ при $\sum_{k:(ij,k)\in E'} s_{ij,k} = c_{ij}$. Если обозначить $y_k := \min_{ij\in k} s_{ij.k}$, то получим в точности двойственную задачу линейного программирования (как?)
- Чем хороша двойственная задача – любое допустимое решение двойственной задачи дает оценку снизу исходной задачи. Например, можно делить веса дуг *ij* поровну.
- Можно брать не все контура, а только часть, и регулировать тем самым число переменных
- Оптимизированную нижнюю оценку можно использовать в алгоритме ветвей и границ
- Более подробно применение техники генерации столбцов посмотрим позже на примере задачи календарно-сетевого планирования

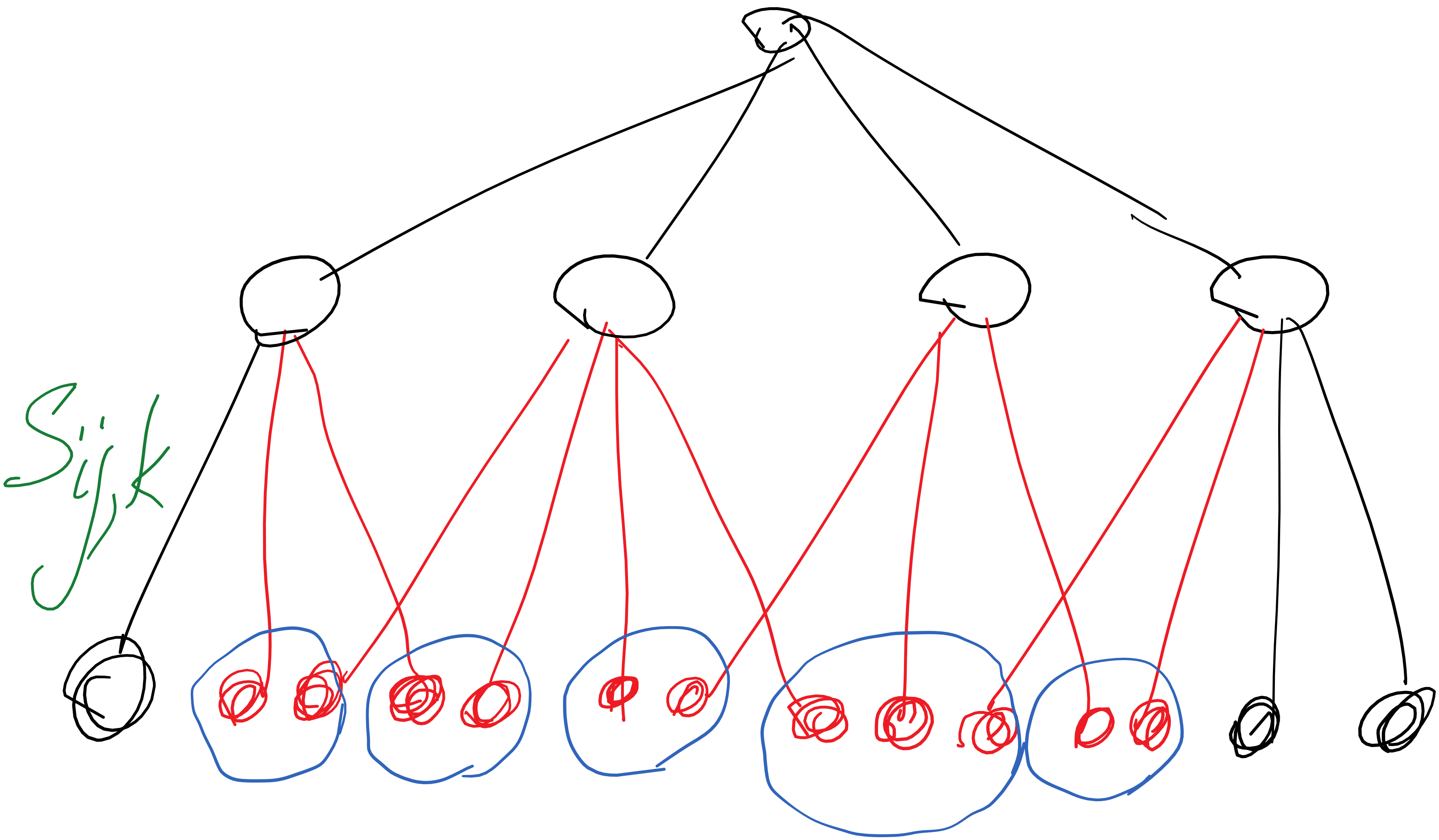
Сетевое представление задачи о покрытии



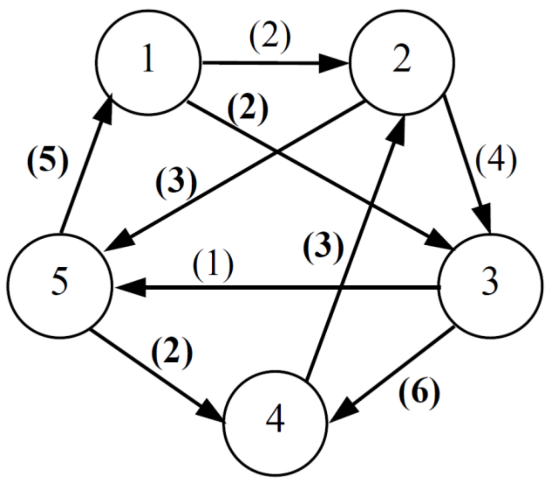
Сетевое представление задачи о покрытии

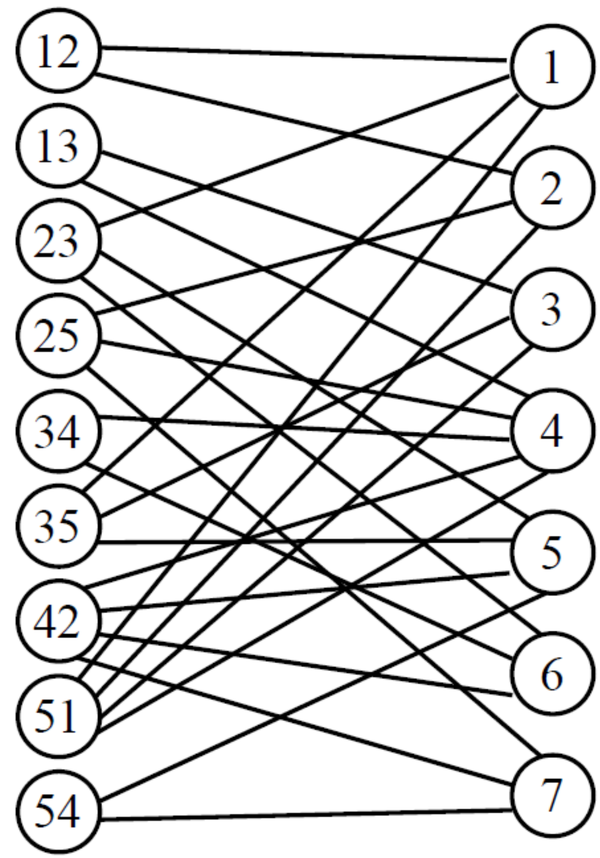


Древовидное представление задачи о покрытии. Оценочная задача



Пример задачи оптимизации последовательности выполнения проектов





Задача о максимальном ациклическом подграфе: другие подходы

- Сводится к квадратичной задаче о назначениях – Quadratic Assignment Problem:

$$\sum c_{ij}\, x_{ik} x_{jl} d_{lk} \to max, \text{ где } d_{lk} := \begin{cases} 1, l > k \\ 0, l \le k \end{cases},$$

$X = (x_{ik})$ – (n x n) (0,1)-матрица назначения, то есть

$x_{ij} \in \{0,1\}, \sum_{i=1}^{n} x_{ij} = 1, j = 1, \dots, n, \sum_{j=1}^{n} x_{ij} = 1, i = 1, \dots, n.$

- Сводится к задаче двоичного линейного программирования – binary linear programming (BLP):

$$\sum c_{ij}\, x_{ij} \to max$$

где $x_{ij} := \begin{cases} 1, \text{проект } i \text{ выполняется до проекта } j \\ 0, \text{в противном случае} \end{cases}$, то есть

$x_{ij} \in \{0,1\},\ x_{ii} := 0, x_{ji} := 1 - x_{ij},\ 0 \le x_{ij} + x_{jk} - x_{ik} \le 1$ для $1 \le i < j < k \le n.$

- Как выглядят непрерывные релаксации для этих задач?

Задача о максимальном ациклическом подграфе

- Первые алгоритмы Lenstra (1973), Kaas (1981) использовали нижние оценки на основе довольно простых эвристик
- I. Charon, O. Hudry / Discrete Applied Mathematics 154 (2006) 2097 – 2116, нижняя оценка на основе лагранжевой релаксации линейной задачи. Лагранжиан:

$$L = \sum_{i,j:i<j} [c_{ij}x_{ij} + c_{ji}(1 - x_{ij}) + \sum\nolimits_{k>j} \lambda_{ijk} \left(x_{ij} + x_{jk} - x_{ik}\right) -$$

$$- \sum\nolimits_{k>j} \mu_{ijk} \left(x_{ij} + x_{jk} - x_{ik} - 1\right)] = \sum\nolimits_{i,j:i<j} [a_{ij} + b_{ij}x_{ij}], \text{ где } a_{ij} \coloneqq c_{ji} + \sum\nolimits_{k:k>j} \mu_{ijk},$$

$$b_{ij} \coloneqq c_{ij} - c_{ji} + \sum\nolimits_{k:k>j} (\lambda_{ijk} - \mu_{ijk}) + \sum\nolimits_{k:k<i} (\lambda_{kij} - \mu_{kij}) - \sum\nolimits_{k:i<k<j} (\lambda_{ikj} - \mu_{ikj}).$$

Каково решение релаксированной задачи?

Подстройка множителей Лагранжа λ_{ijk}, μ_{ijk}:

$$\lambda_{ijk}(t+1) = \max[0; \lambda_{ijk}(t) + \gamma(t) \cdot (x_{ij}(t) + x_{jk}(t) - x_{ik}(t))],$$

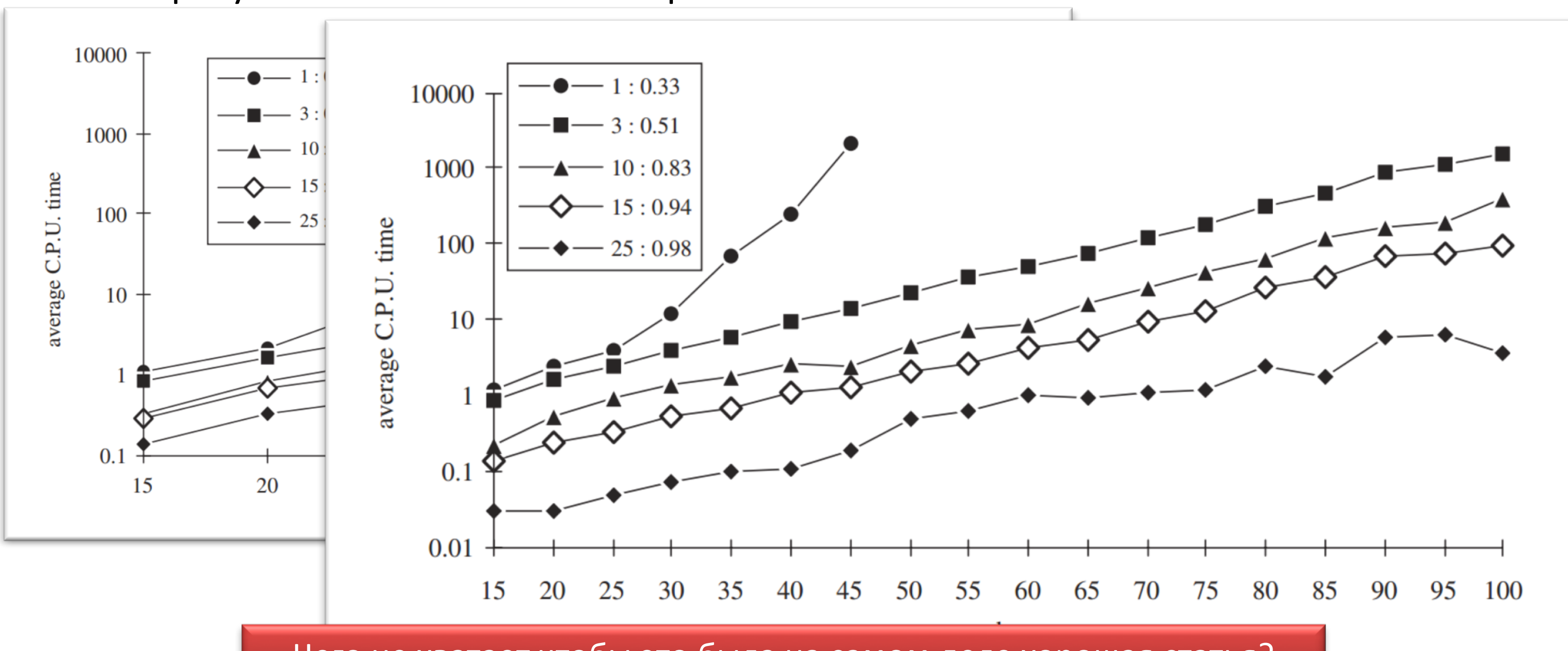
$$\mu_{ijk}(t+1) = \max\left[0; \mu_{ijk}(t) + \gamma(t) \cdot \left(x_{ij}(t) + x_{jk}(t) - x_{ik}(t) - 1\right)\right].$$

Схема ветвления?

Как вычисляется рекорд ?

Задача о максимальном ациклическом подграфе

- I. Charon, O. Hudry / Discrete Applied Mathematics 154 (2006) 2097 – 2116
- Еще 10 страниц улучшений и уточнений…
- и результаты численных экспериментов:



Чего не хватает чтобы это была на самом деле хорошая статья?