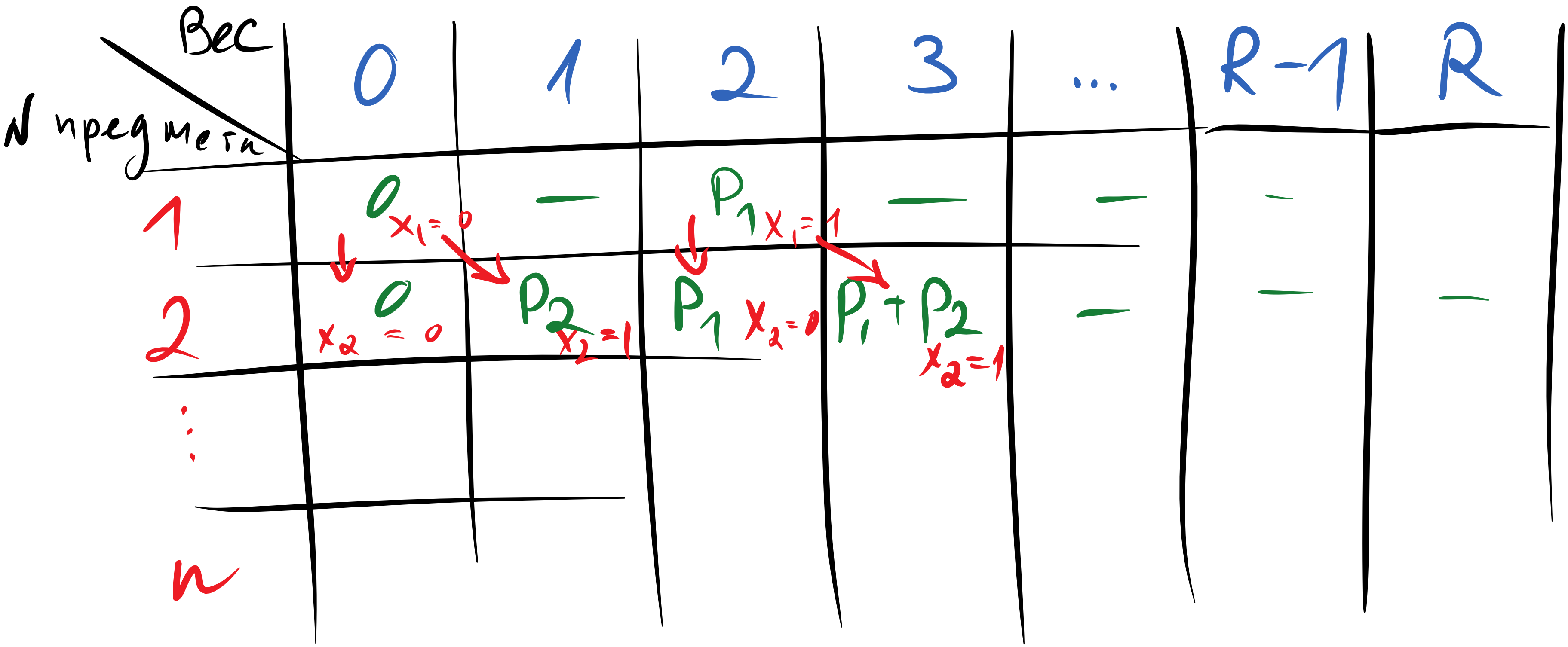
Лекция 4. Введение в дихотомическое и сетевое программирование

Дихотомическое представление функций и ограничений. Линейная задача дискретного программирования. Решение задачи о ранце методом дихотомического программирования. Сетевое программирование и двойственные задачи

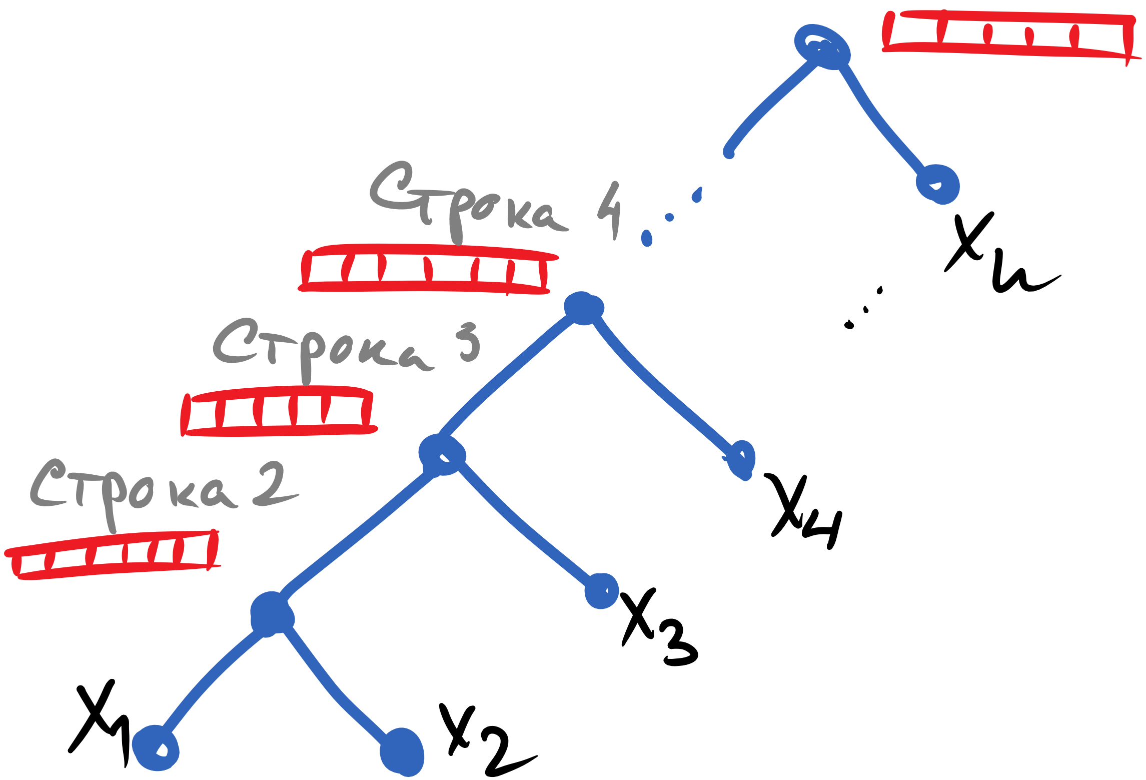
Динамическое и дихотомическое программирование

На одной из предыдущих лекций мы рассматривали метод решения задач дискретной оптимизации с помощью динамического программирования. При решении задачи о ранце строилась такая таблица:



Динамическое и дихотомическое программирование

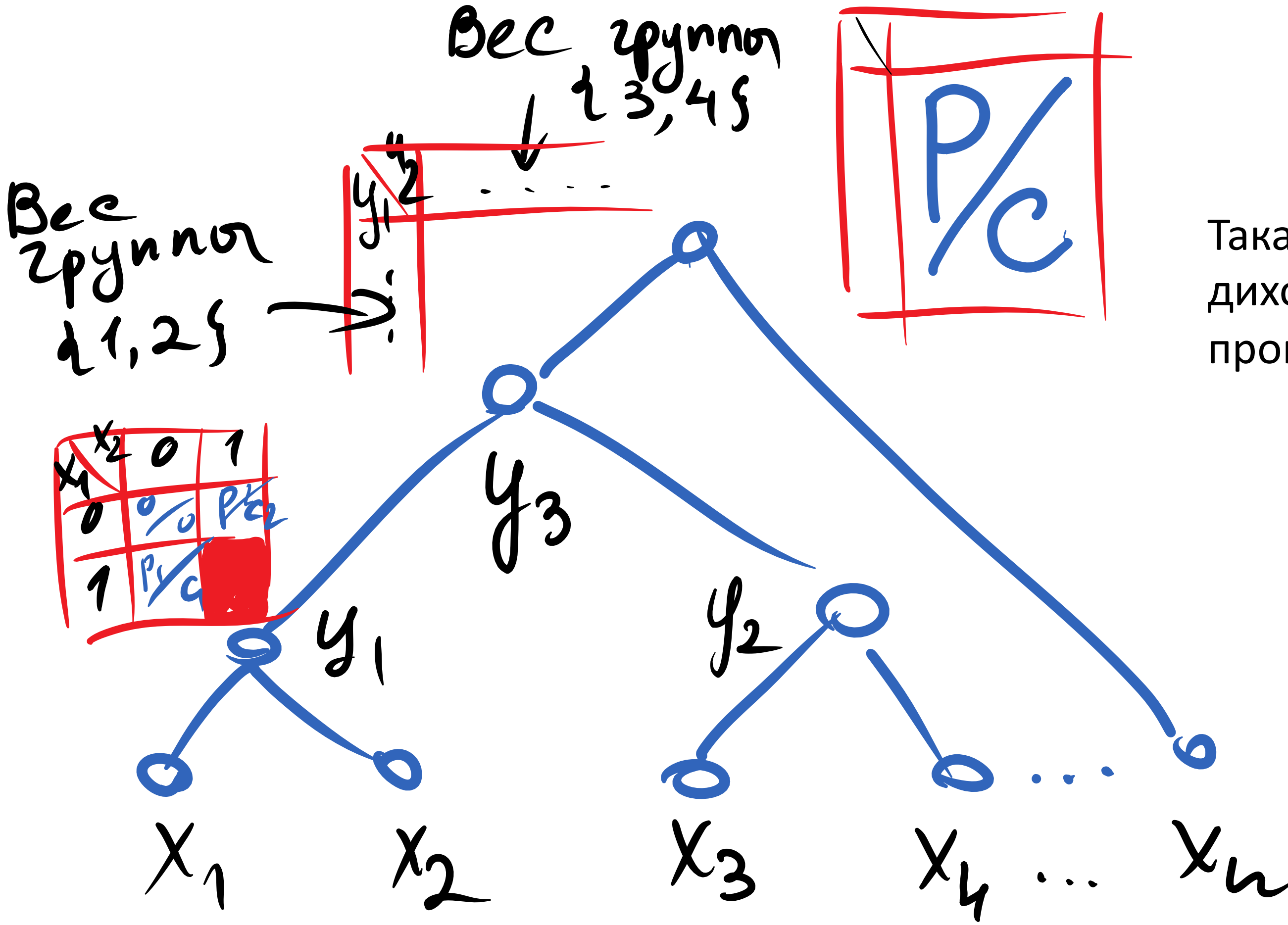
i-я строка – это решение задачи о ранце для множества предметов {1,…,*i*}.
На каждом шаге добавляется один объект.
Процесс построения таблицы можно изобразить так:



Получившееся «дерево» называется беллмановской ветвью

Динамическое и дихотомическое программирование

А почему не рассмотреть более сложные схемы объединения переменных?



Такая схема называется дихотомическим программированием

Общая схема дихотомического программирования

Многие задачи дискретной оптимизации сводятся к следующей постановке: определить вектор $x = \{x_i\}$ с дискретными компонентами, минимизирующий аддитивную функцию

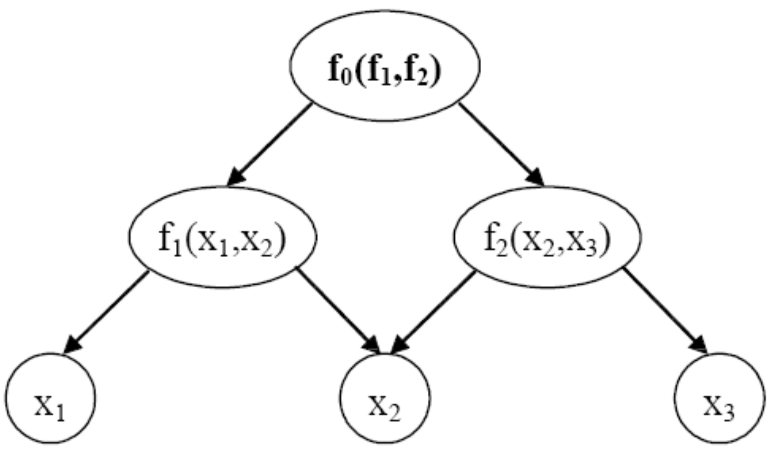
$$\varphi(x) = \sum_{i=1}^{n} \varphi_i(x_i)$$

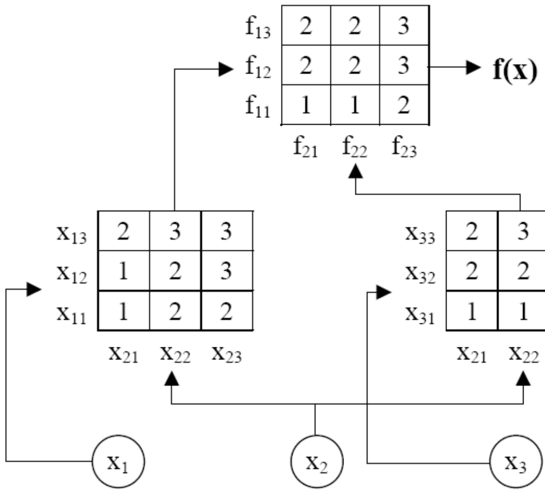
при ограничении

$$f(x) \geq b.$$

Как записать в дихотомическом виде систему неравенств?

Дихотомическое представление функции $f(x) = f_0[f_1(x_1,x_2), f_2(x_2,x_3)]$



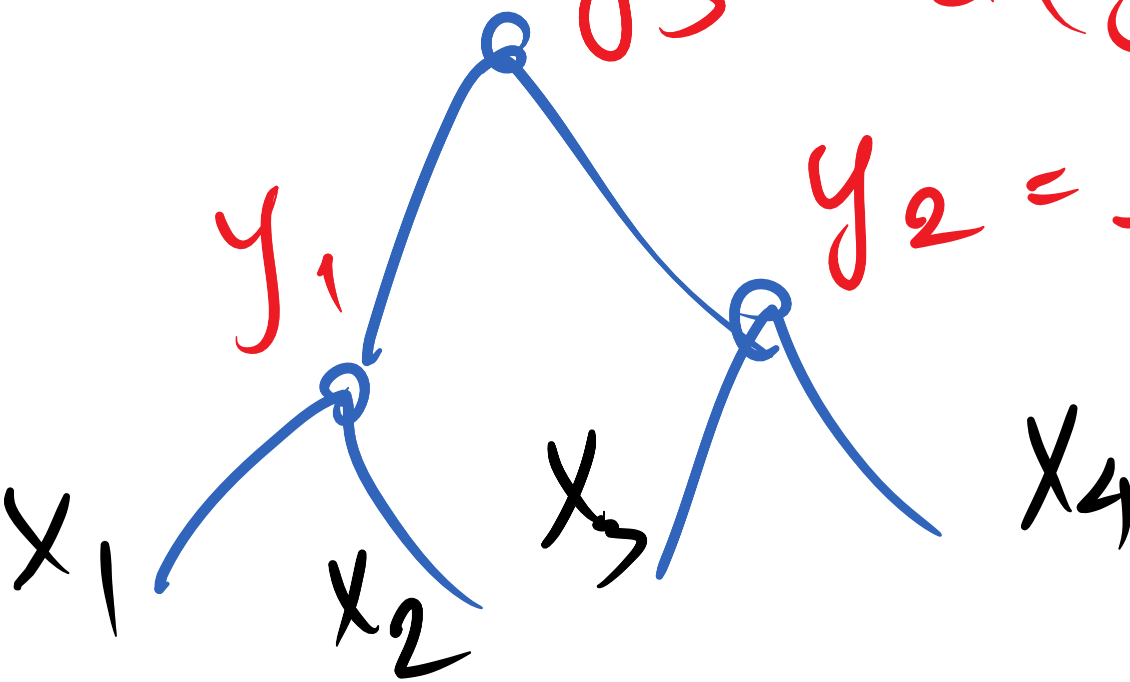


Дихотомическое представление функций и переменных

Дихотомическое представление типа дерева

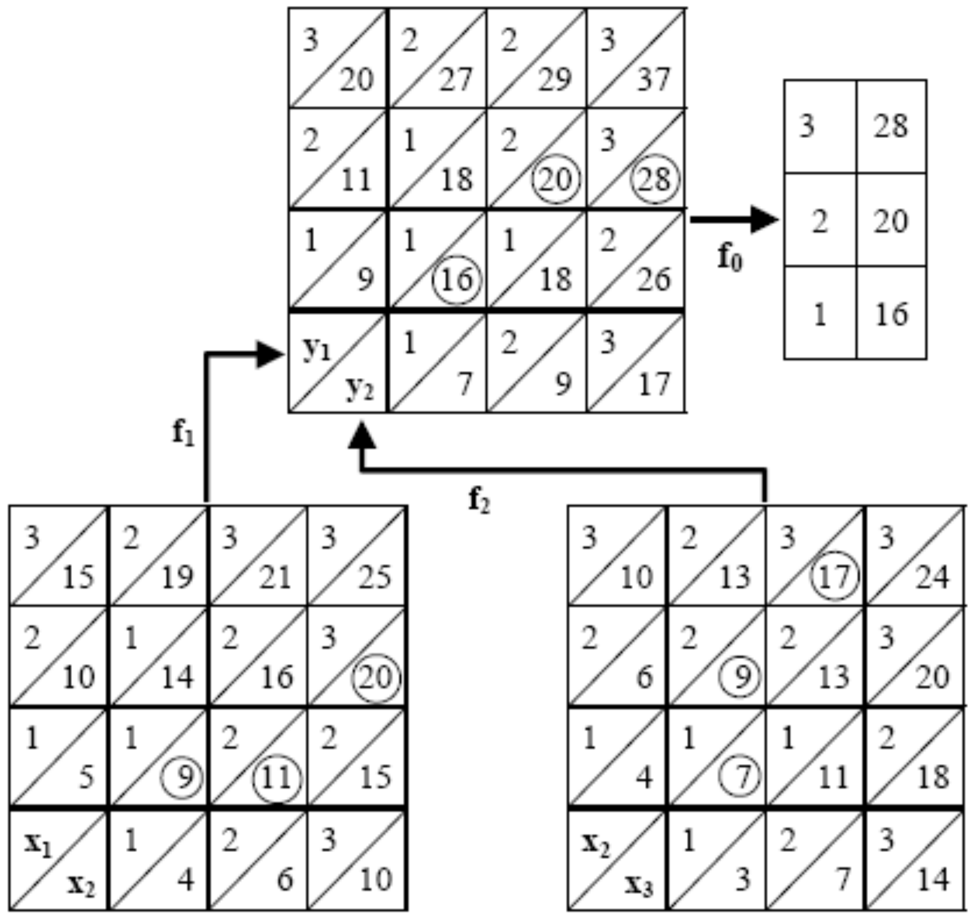
$$y_3 = f_3(y_1, y_2)$$

$$y_2 = f_2(x_3, x_4)$$



Решение задачи методом дихотомического программирования

Если функция ограничений имеет представление типа дерева, задача эффективно решается

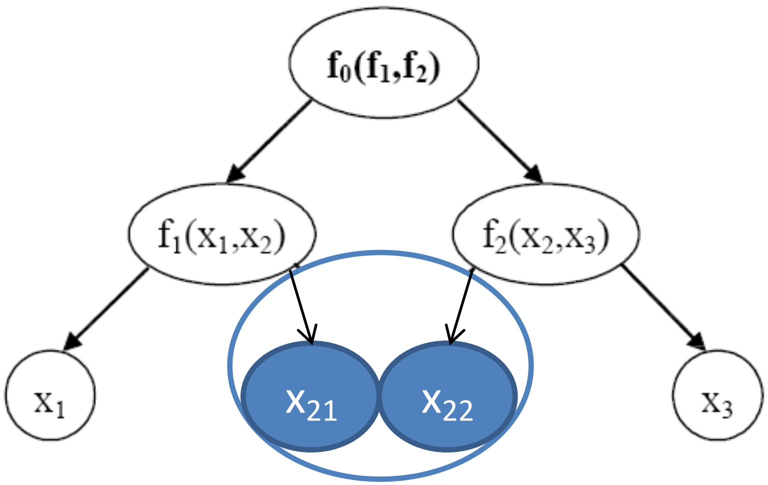


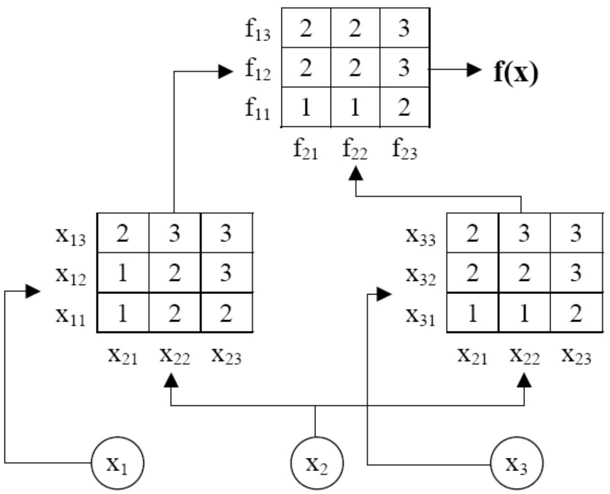
Сетевое программирование

Решаем задачу $\varphi(x) = \varphi_1(x_1) + \ldots + \varphi_n(x_n) \to \min$

при $f(x) \geq b$

Что же делать если функция $f(x)$ не имеет представления типа дерева?





Каждый раз когда переменная x_i используется в дихотомическом представлении k раз, она разбивается на k переменных $x_{i1}, \ldots, x_{ik}$. Соответственно и затраты $\varphi_i(x_i)$ заменяются на $\varphi_{i1}(x_{i1}) + \ldots + \varphi_{ik}(x_{ik})$ таким образом, что для любого x_i верно равенство $\varphi_i(x_i) = \varphi_{i1}(x_i) + \ldots + \varphi_{ik}(x_i)$.

Сетевое программирование

Задача P':

$\varphi'(x') = \varphi_1(x_1) + \varphi_{i\text{-}1}(x_{i\text{-}1}) + \varphi_{i1}(x_i) + \ldots + \varphi_{ik}(x_i) + \varphi_{i+1}(x_{i+1}) + \ldots + \varphi_n(x_n)$ -> min

при $f'(x') \geq b$

решается методом дихотомического программирования.

Но как ее решение соотносится с решением исходной задачи?

Теорема. Решение задачи P‘ дает нижнюю оценку решения исходной задачи P.

Задача наилучшей нижней оценки сводится к поиску наилучшего разбиения переменных.

Эту нижнюю оценку можно использовать, например, в методе ветвей и границ.