Смешанное целочисленное программирование
Mixed Integer Programming (MIP)

Зачем изучаем смешанное целочисленное программирование

- Лучший практический в общем случае метод для оптимизации в целочисленных переменных
- Сильно отличается от ранее рассмотренных методов тем, что использует линейную алгебру и математические свойства задачи
- Гарантирует глобальную оптимальность и наиболее эффективный в общем случае подход к оптимизации
- Рассмотрим уже полюбившееся задачи в рамках новой парадигмы

Смешанное целочисленное программирование

Линейное программирование

Линейное программирование

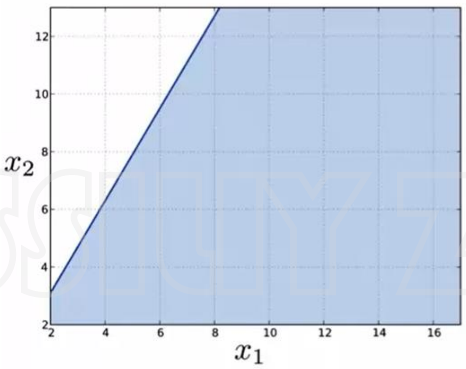
$$\min \quad c_1x_1 + \dots + c_nx_n$$

При этом
$$a_{11}x_1 + \dots + a_{1n}x_n \le b_1$$
$$\dots$$
$$a_{m1}x_1 + \dots + a_{mn}x_n \le b_m$$

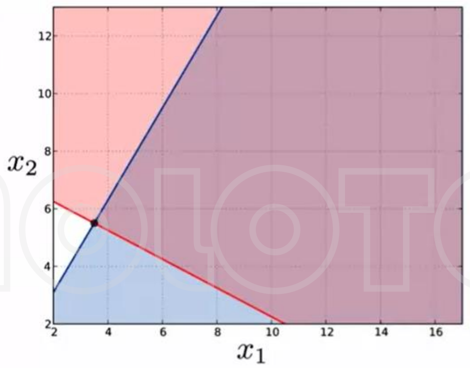
$$x_i \ge 0$$

- n переменных
- m ограничений
- Переменные неотрицательные
- Ограничения определяют неравенства
- Решается за полиномиальное время – симплекс метод, двойственный метод и т.д.

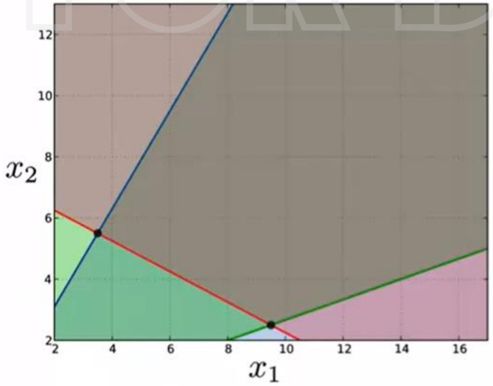
Гиперплоскости



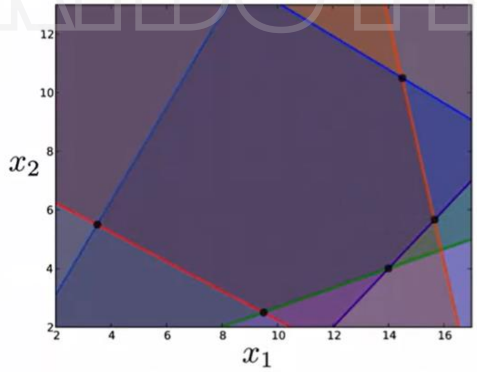
$2.5x_1 \; + \; 4.0x_2 \; \leq \; -0.25$



$-6.0x_1 \; + \; 3.0x_2 \; \leq \; -43.50$

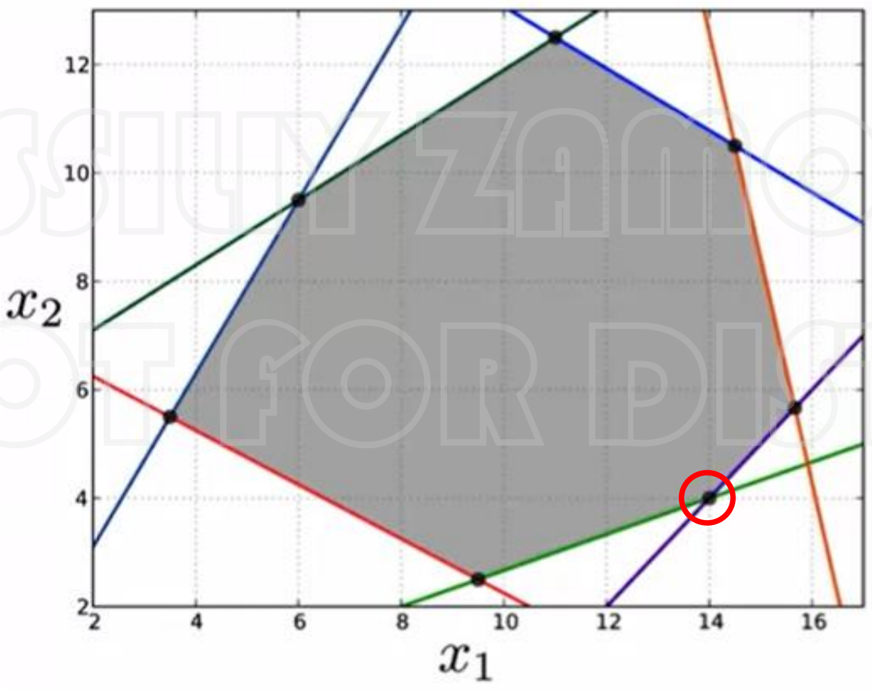


$-4.5x_1 \; - \; 1.5x_2 \; \geq \; 3.00$



$3.5x_1 \; - \; 2.0x_2 \; \geq \; 65.75$

Гиперплоскости



$$
\begin{array}{rcrcr}
2.5x_1 & + & 4.0x_2 & \leq & -0.25 \\
-6.0x_1 & + & 3.0x_2 & \leq & -43.50 \\
-4.5x_1 & - & 1.5x_2 & \geq & 3.00 \\
-1.6x_1 & - & 1.6x_2 & \geq & 16.66 \\
1.2x_1 & + & -4.8x_2 & \geq & 82.33 \\
3.5x_1 & - & 2.0x_2 & \geq & 65.75 \\
5.0x_1 & + & 3.0x_2 & \leq & 29.50
\end{array}
$$

Линейное программирование

$$\min \quad c_1x_1 + \dots + c_nx_n$$

При этом

$$a_{11}x_1 + \dots + a_{1n}x_n \le b_1$$

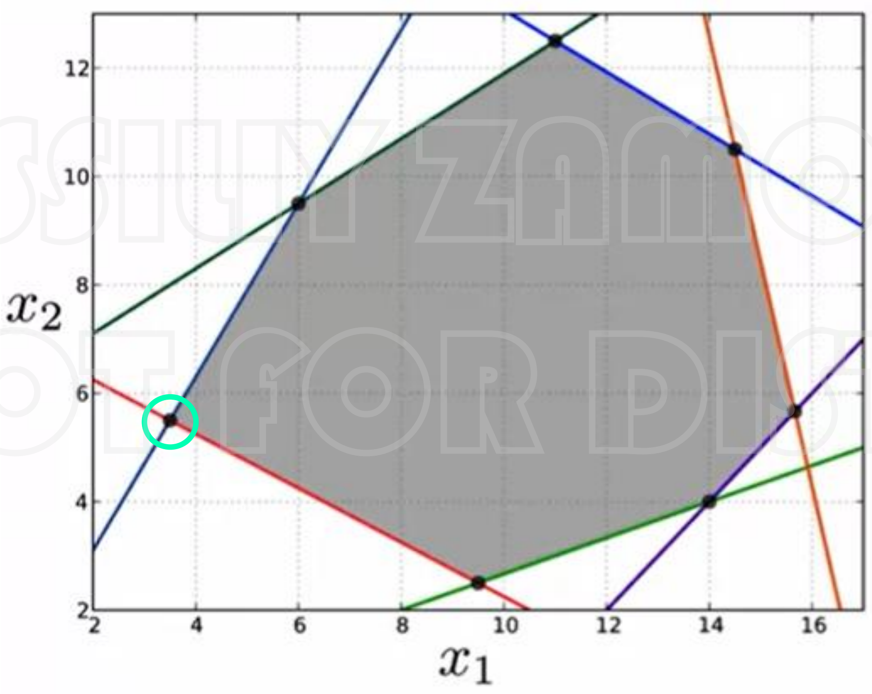
$$\dots$$

$$a_{m1}x_1 + \dots + a_{mn}x_n \le b_m$$

$$x_i \ge 0$$

- Теорема: Как минимум одна точка, где целевая функция достигает минимума находится в вершине допустимого множества

Гиперплоскости



$$
\begin{array}{rcrcr}
2.5x_1 & + & 4.0x_2 & \leq & -0.25 \\
-6.0x_1 & + & 3.0x_2 & \leq & -43.50 \\
-4.5x_1 & - & 1.5x_2 & \geq & 3.00 \\
-1.6x_1 & - & 1.6x_2 & \geq & 16.66 \\
1.2x_1 & + & -4.8x_2 & \geq & 82.33 \\
3.5x_1 & - & 2.0x_2 & \geq & 65.75 \\
5.0x_1 & + & 3.0x_2 & \leq & 29.50
\end{array}
$$

Базисное решение

$$a_{11}x_1 + \cdots + a_{1n}x_n = b_1$$

$$\ldots$$

$$a_{m1}x_1 + \cdots + a_{mn}x_n = b_m$$

$$x_i \geq 0$$

- Попробуем начать с системы линейных уравнений, для описания многогранника
- Решать такие задачи мы умеем

Базисное решение

$$x_1 = b_1 + \sum_{i=m+1}^{n} a_{1i}x_i$$

$$\ldots$$

$$x_m = b_m + \sum_{i=m+1}^{n} a_{mi}x_i$$

$$x_i \geq 0$$

- Решение будет заключаться в выражении базисных переменных через небазисные

Решение системы неравенств

$$a_{11}x_1 + \dots + a_{1n}x_n \le b_1$$

$$\dots$$

$$a_{m1}x_1 + \dots + a_{mn}x_n \le b_m$$

$$\boxed{\frac{n!}{m!(n-m)!}}$$

$$a_{11}x_1 + \dots + a_{1n}x_n + s_1 = b_1$$

$$\dots$$

$$a_{m1}x_1 + \dots + a_{mn}x_n + s_m = b_m$$

$$s_i \ge 0$$

Пример

$$
\begin{array}{rcrcrcrcrcrcrcl}
3x_1 & - & 2x_2 & + & x_3 & & & & & & & = & 1 \\
2x_1 & & & & & + & x_4 & & & + & x_6 & = & 2 \\
x_1 & & & & & & & + & x_5 & + & x_6 & = & 3
\end{array}
$$

$$
\boxed{
\begin{array}{rcrcrcrcr}
x_3 & = & 1 & - & 3x_1 & - & 2x_2 & & \\
x_4 & = & 2 & - & 2x_1 & & & + & x_6 \\
x_5 & = & 3 & - & x_1 & & & + & x_6
\end{array}
}
$$

Пример

$$\begin{array}{lcrcrcrcr} x_3 & = & 1 & - & 3x_1 & - & 2x_2 & & \\ x_4 & = & 2 & - & 2x_1 & & & + & x_6 \\ x_5 & = & 3 & - & x_1 & & & + & x_6 \end{array}$$

$$\downarrow$$

$$\begin{array}{lcrcrcrcr} x_2 & = & \frac{1}{2} & - & \frac{3}{2}x_1 & - & \frac{1}{2}x_3 & & \\ x_4 & = & 2 & - & 2x_1 & & & + & x_6 \\ x_5 & = & 3 & - & x_1 & & & + & x_6 \end{array}$$

$$\begin{array}{lcrcrcrcr} x_3 & = & 1 & - & 3x_1 & - & 2x_2 & & \\ x_4 & = & 2 & - & 2x_1 & & & + & x_6 \\ x_5 & = & 3 & - & x_1 & & & + & x_6 \end{array}$$

$$\downarrow$$

$$\begin{array}{lcrcrcrcrcr} x_3 & = & -8 & - & 2x_2 & + & 3x_5 & - & 3x_6 \\ x_4 & = & -4 & & & + & 2x_5 & - & x_6 \\ x_1 & = & 3 & & & - & x_5 & + & x_6 \end{array}$$

Пример

$$
\begin{array}{rclrrl}
x_3 & = & 1 & -3x_1 & -2x_2 & \\
x_4 & = & 2 & -2x_1 & & +x_6 \\
x_5 & = & 3 & -x_1 & & +x_6
\end{array}
\qquad
\boxed{\begin{array}{c} \frac{1}{3} \\ 1 \\ 3 \end{array}}
$$

$$
l = \underset{i:a_{ie}<0}{\text{arg-min}} \frac{b_i}{-a_{ie}}
$$

$$
\begin{array}{rcrrrr}
x_1 & = & \frac{1}{3} & -\frac{2}{3}x_2 & -\frac{1}{3}x_3 & \\
x_4 & = & \frac{4}{3} & +\frac{4}{3}x_2 & +\frac{2}{3}x_3 & +x_6 \\
x_5 & = & \frac{8}{3} & +\frac{2}{3}x_2 & +\frac{1}{3}x_3 & +x_6
\end{array}
$$

Симплекс метод

- Когда можно остановиться?
- Базисное решение оптимально тогда, когда целевая функция выражена в базисных переменных с неотрицательными коэффициентами:

$$c_0 + c_1 x_1 + \ldots + c_n x_n$$

$$c_i \geq 0 \ \ (1 \leq i \leq n).$$

Симплекс метод

$$
\begin{array}{lrcrcrcrcrcl}
\min & x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & & \\
\text{subject to} & & & & & & & & & & & \\
& 3x_1 & + & 2x_2 & + & x_3 & & & & & = & 1 \\
& 5x_1 & + & x_2 & + & x_3 & + & x_4 & & & = & 3 \\
& 2x_1 & + & 5x_2 & + & x_3 & & & + & x_5 & = & 4
\end{array}
$$

$$\downarrow$$

$$
\begin{array}{lrcrcrcr}
\min & & & \boxed{6 - 3x_1 - 3x_2} & & & & \\
\text{subject to} & & & & & & & \\
& x_3 & = & 1 & - & 3x_1 & - & 2x_2 \\
& x_4 & = & 2 & - & 2x_1 & + & x_2 \\
& x_5 & = & 3 & + & x_1 & - & 3x_2
\end{array}
$$

Оптимальное решение

$$
\begin{array}{lrcrcrcr}
\min & & & 6 & - & 3x_1 & - & 3x_2 \\
\text{subject to} & & & & & & & \\
 & x_3 & = & 1 & - & 3x_1 & - & 2x_2 \\
 & x_4 & = & 2 & - & 2x_1 & + & x_2 \\
 & x_5 & = & 3 & + & x_1 & - & 3x_2
\end{array}
$$

$$\downarrow$$

$$
\begin{array}{lrcrcrcr}
\min & & & \frac{9}{2} & + & \frac{3}{2}x_1 & + & \frac{3}{2}x_3 \\
\text{subject to} & & & & & & & \\
 & x_2 & = & \frac{1}{2} & - & \frac{3}{2}x_1 & - & \frac{1}{2}x_3 \\
 & x_4 & = & \frac{5}{2} & - & \frac{7}{2}x_1 & - & \frac{1}{2}x_3 \\
 & x_5 & = & \frac{3}{2} & + & \frac{11}{2}x_1 & + & \frac{3}{2}x_3
\end{array}
$$

Смешанное целочисленное программирование

Целочисленное программирование

Целочисленное программирование

$$\max \quad c_1x_1 \quad + \cdots + c_nx_n$$

При этом $$a_{11}x_1 + \cdots + a_{1n}x_n \le b_1$$

$$\cdots$$

$$a_{m1}x_1 + \cdots + a_{mn}x_n \le b_m$$

$x_i \geq 0, \quad x_i$ целые

- n переменных
- m ограничений
- Переменные неотрицательные и целые
- Ограничения определяют неравенства
- Наличие целочисленности переводит задачу из класса P в NP

Задача наполнения рюкзака

1.3 кг 2500$

0.25 кг 1100$

0.25 кг 1100$

1.6 кг 1900$

0.48 кг 1200$

0.04 кг 430$

0.04 кг 430$

Вместимость
рюкзака 1.8 кг

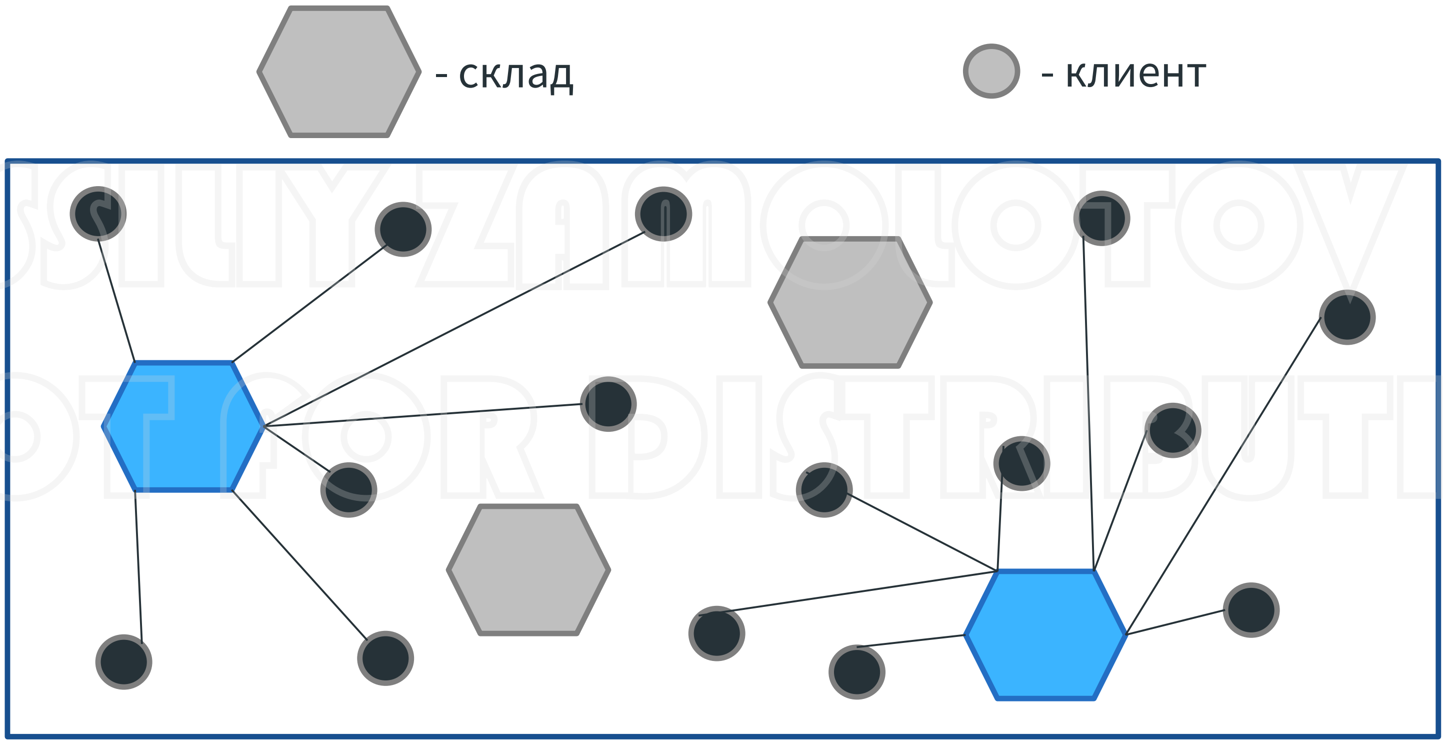
Задача наполнения рюкзака

Maximize $\sum_{i\in 1\ldots j} v_i x_i$

При этом, $\sum_{i\in 1\ldots j} w_i x_i \leq k,\ x_i \in \{0,1\}, (i \in 1 \ldots j)$

- Является задачей целочисленного программирования

Расположение складов



- Необходимо открыть склады в таких местах, чтобы минимизировать расходы на содержание складов и на транспортировку клиентам

Расположение складов

- Как мы можем представить эту задачу в качестве задачи целочисленного программирования?
  - Что будет в качестве переменных?
  - Что будет в качестве ограничений?
  - Что будет целевой функцией?
- Переменные
  - Для каждого склада бинарная переменная x_w, $x_w = 1$ если склад w открыт
  - Для каждого клиента бинарная переменная y_{wc}, $y_{wc} = 1$ если клиент c обслуживается складом w

Расположение складов

- Переменные
  - Для каждого склада бинарная переменная x_w, $x_w = 1$ если склад w открыт
  - Для каждого клиента бинарная переменная y_{wc}, $y_{wc} = 1$ если клиент с обслуживается складом w
- Ограничения
  - Склад может обслужить клиента, только если он открыт: $y_{wc} \leq x_w$
  - Клиент должен обслуживать одним и только одним складом $\sum_{w \in W} y_{wc} = 1$

Расположение складов

- Переменные
  - Для каждого склада бинарная переменная x_w, $x_w = 1$ если склад w открыт
  - Для каждого клиента бинарная переменная y_{wc}, $y_{wc} = 1$ если клиент с обслуживается складом w
- Целевая функция
  - Minimize $\sum_{w\in W} c_w x_w + \sum_{w\in W, c\in C} t_{w,c}\, y_{w,c}$, где c_w - фиксированные расходы на содержание склада, $t_{w,c}$ - расходы на доставку от склада w до клиента c.

Расположение складов

$$Minimize \sum_{w \in W} c_w x_w + \sum_{w \in W, c \in C} t_{w,c}\, y_{w,c}$$

При этом $y_{wc} \le x_w \ (w \in W, c \in C)$

$$\sum_{w \in W} y_{wc} = 1 \ (c \in C)$$

$$x_w \in \{0,1\} \ (w \in W)$$

$$y_{wc} \in \{0,1\} \ (w \in W, c \in C)$$

Расположение складов

- Переменные
  - Для каждого склада бинарная переменная x_w, $x_w = 1$ если склад w открыт
  - Для каждого клиента бинарная переменная y_{wc}, $y_{wc} = 1$ если клиент c обслуживается складом w
- Почему не используем
  - y_c как целое число с номером склада для клиента c?

Бинарные переменные

- Смешанное целочисленное программирование любит бинарные переменные
  - Целочисленные переменные часто бинарны
- Легко задавать линейные ограничения
  - Когда используются бинарные переменные
- По прежнему много вариантов в моделировании
  - Переменные, ограничения, целевая функция может задаваться по-разному

Метод ветвей и границ

- Как решить задачу смешанного целочисленного программирования?
  - Активно исследовать область решения
- Метод ветвей и границ
  - Ограничение – поиск оптимистично релаксации
  - Ветвление – разбиение задачи на подзадачи
- Смешанное целочисленное программирование имеет натуральную релаксацию
  - Линейная релаксация
  - Удаление целочисленности в переменных

Метод ветвей и границ для СЦП

- Решаем линейную релаксацию
  - Если линейная релаксация хуже, чем лучшее уже найденное решение, отрезаем часть области
  - Если линейная релаксация имеет целочисленное решение, мы имеем выполнимое решение
  - В противном случае, ищем целое x такое, что $x \geq \lceil f \rceil$ и $x \leq \lfloor f \rfloor$, где f – дробное решение линейной релаксации, т.е. повторяем алгоритм с двумя подзадачами

Метод ветвей и границ для СЦП

- Фокус на целевую функцию
  - Релаксация даёт оптимистичное ограничение
- Отсечение базируется на неоптимальности
  - Отсекаем заведомо неоптимальные узлы
- Релаксация выполнимости
  - Релаксация на целочисленность
- Глобальный взгляд на релаксацию
  - Рассматриваем все ограничения задачи

Задача наполнения рюкзака

Maximize $\sum_{i \in 1 \dots j} v_i x_i$

При этом, $\sum_{i \in 1 \dots j} w_i x_i \leq k,\ 0 \leq x_i \leq, 1\ (i \in 1 \dots j)$

i	V_i	W_i
1	45	5
2	48	8
3	35	3

K = 10

Метод ветвей и границ для СЦП

- Линейная релаксация
  - Похожа на жадный алгоритм
- Как происходит ветвление?
  - Переменные с дробными значениями
  - Самый ценный предмет не может быть помещён полностью
- Что означают подзадачи?
  - Берём или не берём тот или иной предмет – что собирается сделать линейная релаксация?

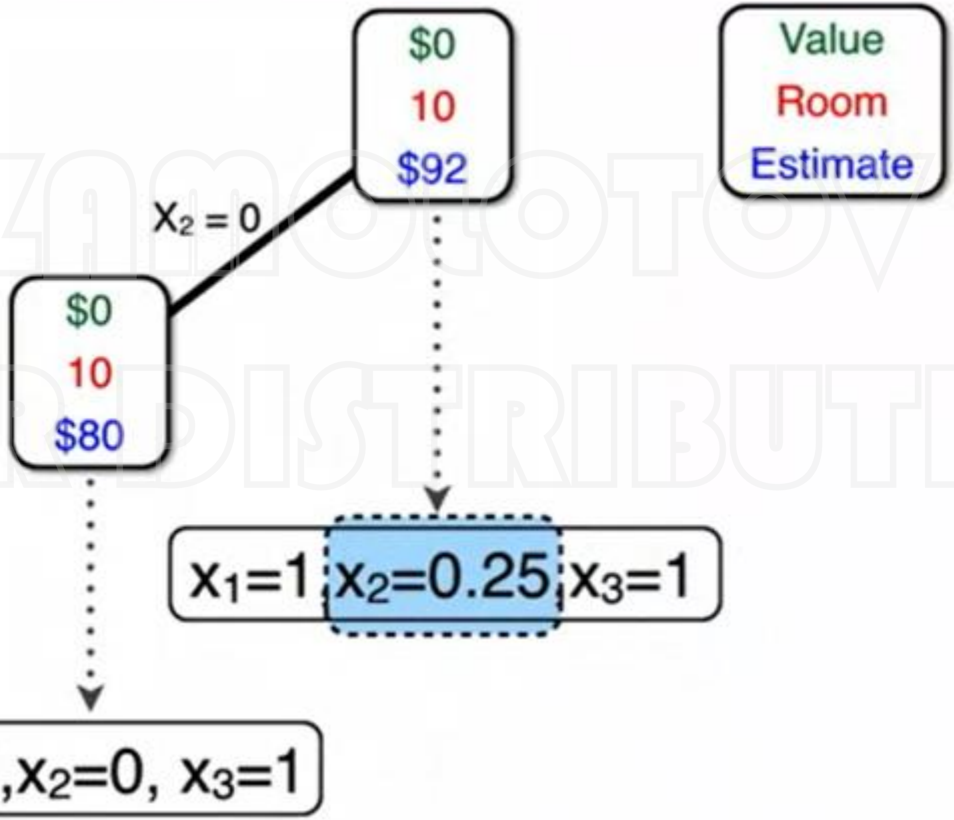
i	V_i	W_i
1	45	5
2	48	8
3	35	3

$K = 10$

Метод ветвей и границ для СЦП

i	V_i	W_i
1	45	5
2	48	8
3	35	3

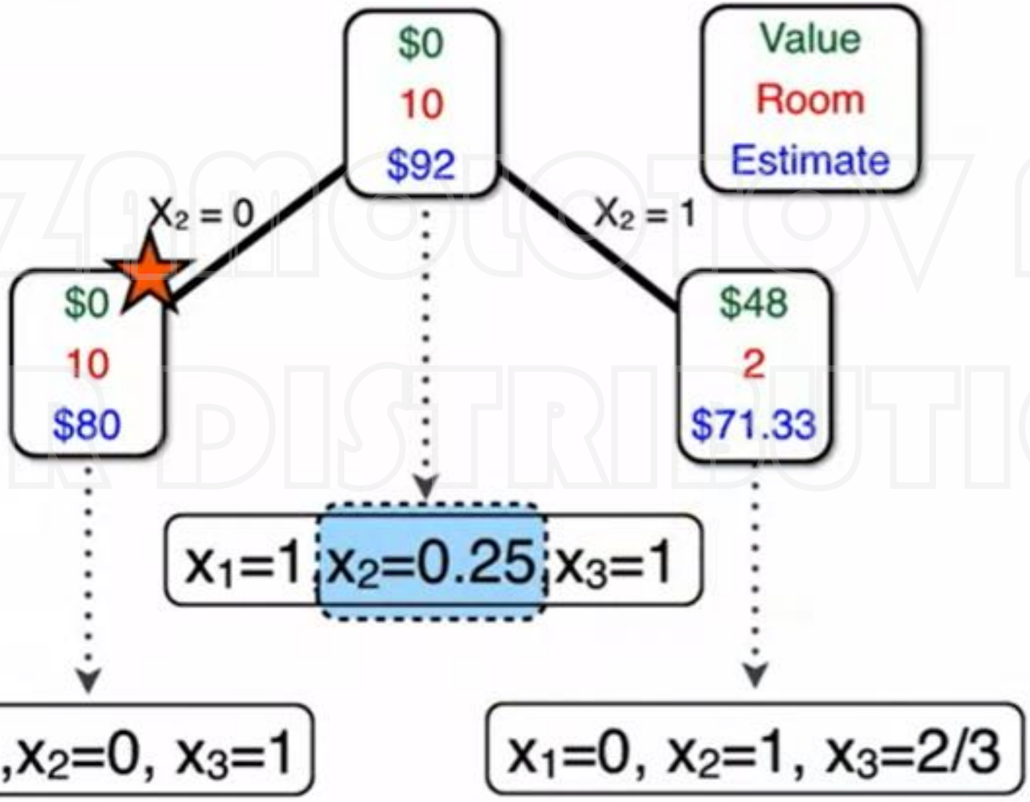
K = 10



Метод ветвей и границ для СЦП

i	V_i	W_i
1	45	5
2	48	8
3	35	3

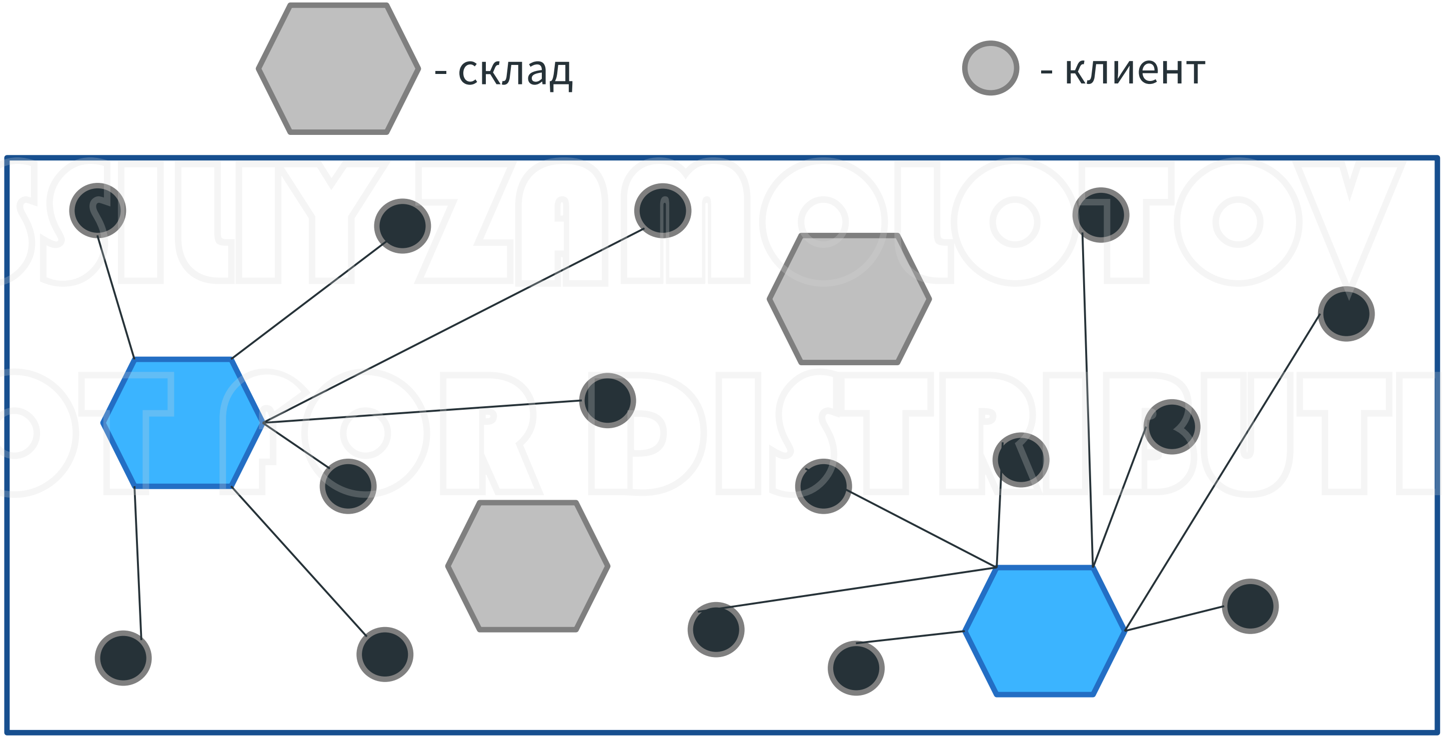
K = 10



Метод ветвей и границ для СЦП

- Когда метод ветвей и границ эффективен?
  - Когда нужно отсечь неоптимальные значения рано
  - Необходимое условие: линейная релаксация сильна
- Что такое хорошая модель СЦП?
  - Модель с хорошей линейной релаксацией

Расположение складов



- Необходимо открыть склады в таких местах, чтобы минимизировать расходы на содержание складов и на транспортировку клиентам

Расположение складов

- Переменные
  - Для каждого склада бинарная переменная x_w, $x_w = 1$ если склад w открыт
  - Для каждого клиента бинарная переменная y_{wc}, $y_{wc} = 1$ если клиент c обслуживается складом w
- Ограничения
  - Склад может обслужить клиента, только если он открыт: $y_{wc} \leq x_w$
  - Клиент должен обслуживать одним и только одним складом $\sum_{w \in W} y_{wc} = 1$

Расположение складов

$$Minimize \sum_{w \in W} c_w x_w + \sum_{w \in W, c \in C} t_{w,c}\, y_{w,c}$$

При этом $\boxed{y_{wc} \le x_w \; (w \in W, c \in C)}$

$$\sum_{w \in W} y_{wc} = 1 \; (c \in C)$$

$$x_w \in \{0,1\} \; (w \in W)$$

$$y_{wc} \in \{0,1\} \; (w \in W, c \in C)$$

Расположение складов

- Переменные
  - Для каждого склада бинарная переменная x_w, $x_w = 1$ если склад w открыт
  - Для каждого клиента бинарная переменная y_{wc}, $y_{wc} = 1$ если клиент c обслуживается складом w
- Ограничения
  - Склад может обслужить клиента, только если он открыт: $y_{wc} \leq x_w$
  - Клиент должен обслуживать одним и только одним складом $\sum_{w \in W} y_{wc} = 1$
- Как записать ограничения иначе?
  - $\sum_{c \in C} y_{wc} \leq |C| x_w$

Расположение складов

$$Minimize \sum_{w \in W} c_w x_w + \sum_{w \in W, c \in C} t_{w,c}\, y_{w,c}$$

При этом

$$\sum_{c \in C} y_{wc} \le |\mathrm{C}| x_w \ (w \in W)$$

$$\sum_{w \in W} y_{wc} = 1 \ (c \in C)$$

$$x_w \in \{0,1\} \ (w \in W)$$

$$y_{wc} \in \{0,1\} \ (w \in W, c \in C)$$

Расположение складов

- Какая из моделей лучше?
  - Новая имеет одно ограничение вместо |C| ограничений для каждого склада
- Рассмотрим в контексте линейной релаксации

Расположение складов

- Решение для
  - $y_{wc} \leq x_w \; (w \in W, c \in C)$
- Это также решение для
  - $\sum_{c \in C} y_{wc} \leq |C| x_w \; (w \in W)$
- Но не наоборот
- Изначальная модель имеет лучшую линейную релаксацию

Расположение складов

- Решение для
  - $y_{wc} \le x_w \ (w \in W, c \in C)$
- Это также решение для
  - $\sum_{c \in C} y_{wc} \le |C| x_w \ (w \in W)$
- Но не наоборот
- Изначальная модель имеет лучшую линейную релаксацию

W	C	Obj1	Obj2	%
16	50	932,615	844,807	9.5
16	50	1,010,541	853,434	15.6
25	50	796,648	659,341	17.2
50	50	793,439	631,421	20.4

Расположение складов

```python
m = gp.Model('warehouse_location')

select = m.addVars(num_facilities, vtype=GRB.BINARY, name='Select')
assign = m.addVars(custumer, vtype=GRB.BINARY, name='Assign')

m.addConstrs((assign[(c,f)] <= select[f] for c,f in cartesian_prod), name='Setup2ship')
m.addConstrs((gp.quicksum(assign[(c,f)] for f in range(num_facilities)) == 1 for c in range(num_customers)), name='Demand')

m.setObjective(select.prod(setup_cost)+assign.prod(shipping_cost), GRB.MINIMIZE)

m.optimize()
```


[Gurobi Python API from Gurobi]

Метод ветвей и границ для СЦП

- Когда метод ветвей и границ эффективен?
  - Когда нужно отсечь неоптимальные значения рано
  - Необходимое условие: линейная релаксация сильна
- Что такое хорошая модель СЦП?
  - Модель с хорошей линейной релаксацией

Раскраска карты

- Постановка:
  - Необходимо раскрасить карту таким образом, чтобы две граничащие территории не имели одинаковый цвет
- Теорема о 4 цветах для раскраски карты:
  - Любая карта может быть раскрашена в 4 цвета
  - Доказана экспериментально

Раскраска карты

```
model = cp_model.CpModel()

color = [
    model.NewIntVar(0, 3, 'color%i' % i) for i in data['region']
]

max_c = [
    model.NewIntVar('max_c')
]

for i in data['region']:
    color[i]<=max_c

for i, j in edges.itertuples(index=False):
    model.Add(color[int(i)]!=color[int(j)])

solver = cp_model.CpSolver()
model.Minimize(max_c)
```


[Python, lib ortools cp solver from Google-OR package](#)

Большое M

- Условие «не равно» не является линейным ограничением
- Можно использовать два неравенства
  - $x \neq y \rightarrow x \leq y - 1\ and\ x \geq y + 1$
- Условие and (дизъюнкция) не поддерживается СЦП
- Введем бинарную переменную b и достаточно большое M, такое что:
  - $x \leq y - 1 + bM$
  - $x \geq y + 1 - (1 - b)M$
- При различных значениях b
  - При b = 1 выполняется ограничение $x \leq y - 1 + bM$ автоматически, а второе ограничение $x \geq y + 1$
  - При b = 0 выполняется ограничение $x \leq y + 1 - (1 - b)M$ автоматически, а второе ограничение $x \leq y - 1$

Большое M

- Как с этим работает линейная релаксация?
  - $x \le y - 1 + bM$
  - $x \ge y + 1 - (1 - b)M$
- Выберем b=0.5
  - Половина большого числа – это все ещё большое число
  - В основном линейная релаксация будет игнорировать эти ограничения

Большое M

$$
\begin{aligned}
obj &\in \{0,1,2,3\} \\
\mathrm{color}_c &\in \{0,1,2,3\} \\
b_{c1,c2} &\in \{0,1\} \\
M &= 4
\end{aligned}
$$

$$
\begin{aligned}
\text{LP: } obj &= 0.0 \\
\mathrm{color} &= 0.0 \\
b &= 0.25
\end{aligned}
$$

$$
\begin{aligned}
&\min && obj \\
&\text{subject to} && \\
& && obj \geq \mathrm{color}_c \quad (c \in C) \\
& && \mathrm{color}_{c_1} \leq \mathrm{color}_{c_2} - 1 + b_{c_1,c_2} M \\
& && \mathrm{color}_{c_1} \geq \mathrm{color}_{c_2} + 1 - (1 - b_{c_1,c_2}) M \\
& && (c_1, c_2 \in C \text{ and adjacent})
\end{aligned}
$$

Нужен как минимум 1 цвет

Оптимальных ветвей 5

Доверительных 65

Раскраска карты

- Сможем ли мы найти модель получше?
- СЦП любит бинарные переменные, поэтому все переменные переведём к 0 и 1
  - Используем бинарные переменные $b_{x=0}$, $b_{x=1}, b_{x=2}, b_{x=3}$
  - $b_{x=i}$ равно 1, если переменная равна $x = i$
- Добавим ограничение по природе переменной, может быть только 1 цвет
  - $b_{x=0} + b_{x=1} + b_{x=2} + b_{x=3} = 1$
- Добавим «не равно» $x \neq y$
  - $b_{x=1} + b_{y=1} \leq 1$
  - $b_{x=2} + b_{y=2} \leq 1$
  - $b_{x=3} + b_{y=3} \leq 1$

Раскраска карты в бинарных переменных

LP:

$$\begin{aligned} obj &= 0.\overline{27} \\ \text{color}_{c,0} &= 0.5 \\ \text{color}_{c,1} &= 0.\overline{27} \\ \text{color}_{c,2} &= 0.1\overline{36} \\ \text{color}_{c,3} &= 0.\overline{09} \end{aligned}$$

Нужно как минимум 2 цвета

$$\begin{array}{lll} obj & \in \{0,1,2,3\} & \\ \text{color}_{c,v} & \in \{0,1\} & \\ \text{min} & obj & \\ \text{subject to} & & \\ & obj \geq v \times \text{color}_{c,v} & (c \in C, v \in 0..3) \\ & \displaystyle\sum_{v=0}^{3} \text{color}_{c,v} = 1 & (c \in C) \\ & \text{color}_{c_1,v} + \text{color}_{c_2,v} \leq 1 & (c_1, c_2 \in C \text{ and adjacent,} \end{array}$$

Оптимальных ветвей 12

Доверительных 22

Раскраска карты в бинарных переменных

- Можно присвоить любой из стран конкретный цвет, это сузит поиск
- Объединить ограничения, это улучшит релаксацию

$$obj \geq v \times \mathrm{color}_{c,v} \quad (c \in C, v \in 0..3)$$

$$\sum_{v=0}^{3} \mathrm{color}_{c,v} = 1 \quad (c \in C)$$

$$\downarrow$$

$$obj \geq \sum_{v=0}^{3} v \times \mathrm{color}_{c,v} \quad (c \in C)$$

Раскраска карты в бинарных переменных

LP: $obj = 0.5$

$color_{c,0} = 0.5$

$color_{c,1} = 0.5$

$color_{c,2} = 0$

$color_{c,3} = 0$

Оптимальных ветвей 9
Доверительных 41

Нужно как минимум 2 цвета

$$
\begin{array}{lll}
obj & \in \{0,1,2,3\} & \\
color_{c,v} & \in \{0,1\} & \\
\min & obj & \\
\text{subject to} & obj \geq \sum_{v=0}^{3} v \times color_{c,v} & (c \in C) \\
 & \sum_{v=0}^{3} color_{c,v} = 1 & (c \in C) \\
 & color_{c_1,v} + color_{c_2,v} \leq 1 & (c_1, c_2 \in C
\end{array}
$$

Смешанное целочисленное программирование

Секущие плоскости, сечение Гомори

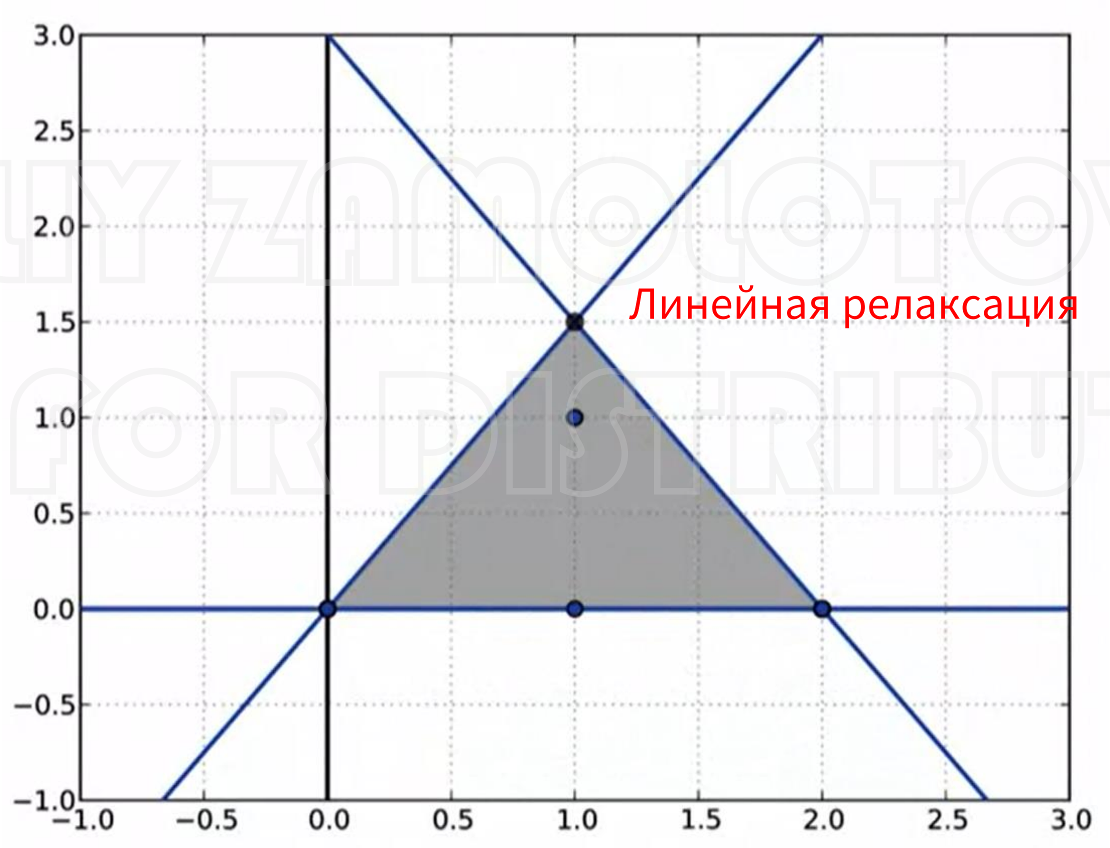
Секущие плоскости

- Добавляем линейное ограничение, такое что:
  - Оно не удаляет допустимых значений
  - Отсекает неоптимальные значения для линейной релаксации

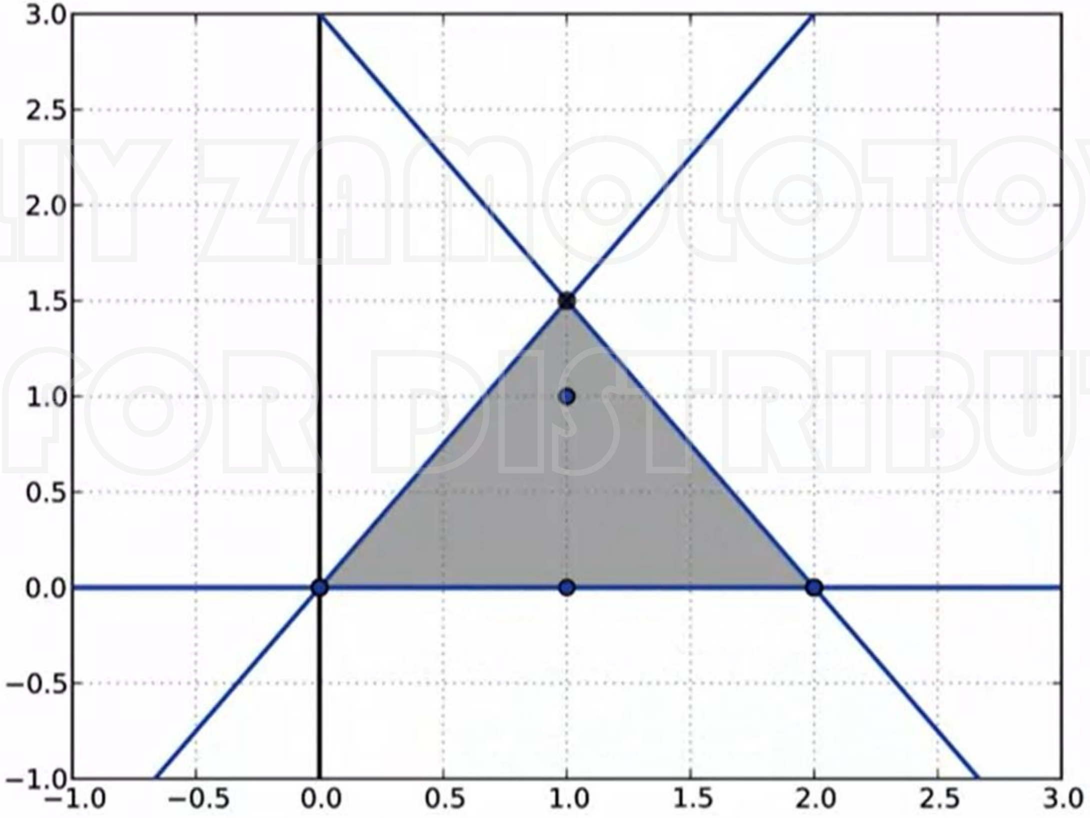
$$\max \quad x_2$$

При этом
$$3x_1 + 2x_2 \leq 6$$
$$-3x_1 + 2x_2 \leq 0$$
$$x_i \geq 0, \ x_i \text{ целые}$$

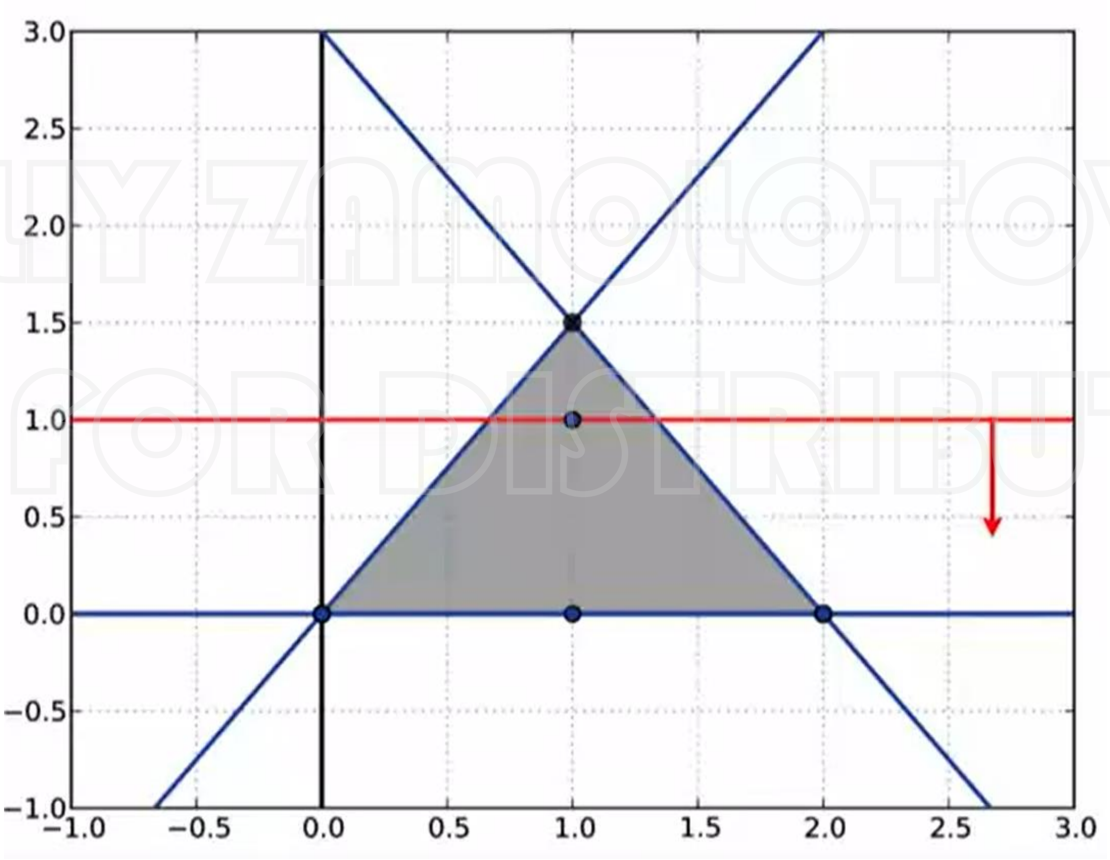
Секущие плоскости



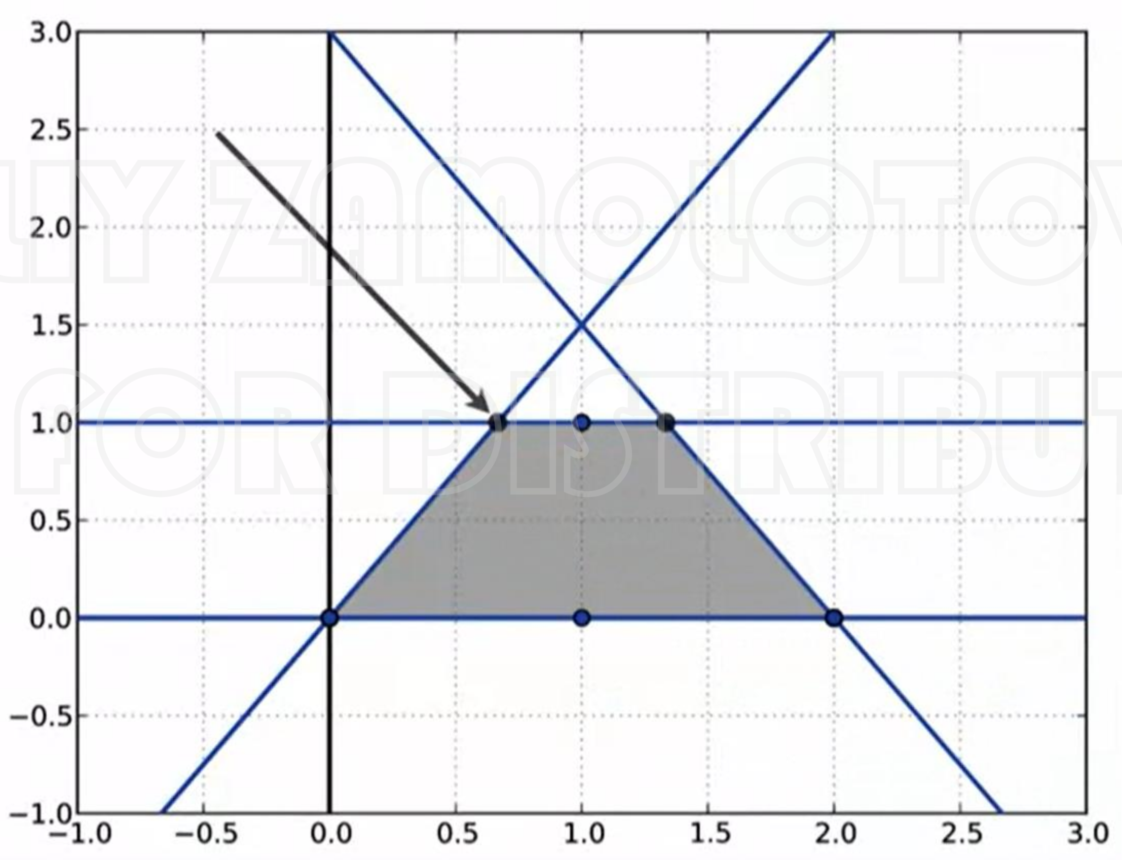
Секущие плоскости



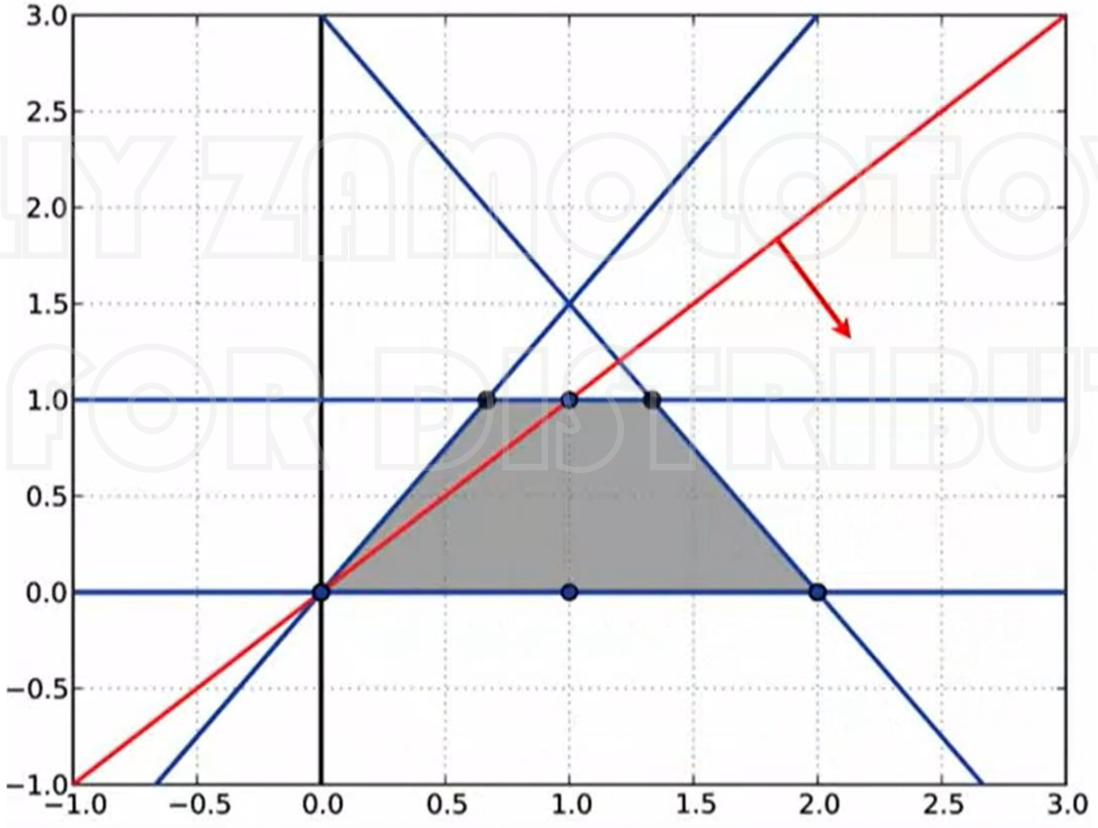
Секущие плоскости



Секущие плоскости



Секущие плоскости



Базовое выполнимое решение

- По решению симплекс метода для линейного программирования:

$$x_1 = b_1 + \sum_{j=m+1} a_{1j} x_j$$

$$\ldots$$

$$x_m = b_m + \sum_{j=m+1} a_{mj} x_j$$

- Базовое выполнимое решение

$$x_1 = b_1$$

$$\ldots$$

$$x_m = b_m$$

$$x_j = 0 \ \ (m < j \leq n)$$

Сечение Гомори

- Допустим, что b_1 - дробное

$$x_1 + \sum_{j=m+1} a_{1j} x_j = b_1$$

$$\sum_{j=m+1} \lfloor a_{1j} \rfloor x_j \leq \sum_{j=m+1} a_{1j} x_j$$

$$\Downarrow$$

$$x_1 + \sum_{j=m+1} \lfloor a_{1j} \rfloor x_j \leq b_1$$

$$x_1 + \sum_{j=m+1} \lfloor a_{1j} \rfloor x_j \ \text{целое}$$

$$\Downarrow$$

$$x_1 + \sum_{j=m+1} \lfloor a_{1j} \rfloor x_j \leq \lfloor b_1 \rfloor$$

Сечение Гомори

$$x_1 + \sum_{j=m+1} \lfloor a_{1j} \rfloor x_j \leq \lfloor b_1 \rfloor$$

- Ограничение не удаляет допустимых значений
- Ограничение отсекает базовые выполнимые значения, которые могут являться оптимальными для линейной релаксации

Сечение Гомори

- Допустим, что b_1 - дробное

$$x_1 + \sum_{j=m+1} a_{1j} x_j = b_1$$

$$-$$

$$x_1 + \sum_{j=m+1} \lfloor a_{1j} \rfloor x_j \leq \lfloor b_1 \rfloor$$

$$=$$

$$\sum_{j=m+1} \left(a_{1j} - \lfloor a_{1j} \rfloor\right) x_j \geq b_1 - \lfloor b_1 \rfloor$$

$$\Downarrow$$

$$\sum_{j=m+1} \left(a_{1j} - \lfloor a_{1j} \rfloor\right) x_j - s = b_1 - \lfloor b_1 \rfloor$$

$$s = \sum_{j=m+1} \left(a_{1j} - \lfloor a_{1j} \rfloor\right) x_j - \left(b_1 - \lfloor b_1 \rfloor\right)$$

Сечение Гомори

- Решаем линейную релаксацию
- Выбираем строку, которая содержит дробную константу и добавляем сечение Гомори
- Применяем двойственный симплекс на обследование выполнимости
- Выполняем пока не найдём целого решения, либо убедимся в выполнимости

Сечение Гомори

$$\max \ x_2$$

При этом
$$3x_1 + 2x_2 \leq 6$$
$$-3x_1 + 2x_2 \leq 0$$
$$x_i \geq 0, \ x_i \text{ целые}$$

$$\min \ -x_2$$

При этом
$$3x_1 + 2x_2 + x_3 = 6$$
$$-3x_1 + 2x_2 + x_4 = 0$$
$$x_i \geq 0, \ x_i \text{ целые}$$

Сечение Гомори

	x_1	x_2	x_3	x_4	b
	0	-1	0	0	0
x_3	3	2	1	0	6
x_4	-3	2	0	1	0

Сечение Гомори

	x_1	x_2	x_3	x_4	b
	0	0	1/4	1/4	3/2
x_1	1	0	1/6	-1/6	1
x_2	0	1	1/4	1/4	3/2

$$\frac{1}{4}x_3 + \frac{1}{4}x_4 \geq \frac{1}{2}$$

- Выразим через x_1 и x_2

$$\min \ -x_2$$

При этом
$$\begin{aligned} 3x_1 + 2x_2 + x_3 &= 6 \\ -3x_1 + 2x_2 + x_4 &= 0 \end{aligned}$$

$$x_i \geq 0, \ x_i \text{ целые}$$

Сечение Гомори

$$\frac{1}{4}x_3 + \frac{1}{4}x_4 \geq \frac{1}{2}$$

$$\min \ -x_2$$

При этом $3x_1 + 2x_2 + x_3 = 6$

$$-3x_1 + 2x_2 + x_4 = 0$$

$$x_i \geq 0, \ x_i \text{ целые}$$

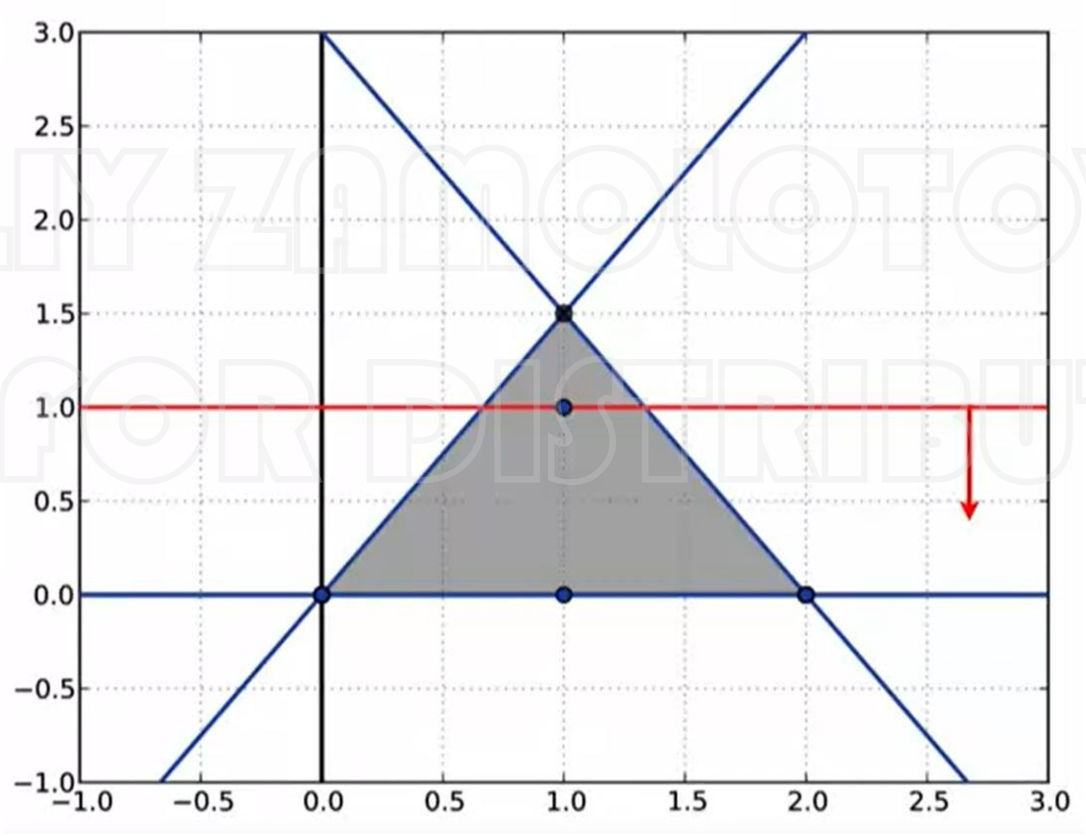
- Выразим через x_1 и x_2

$$\frac{1}{4}(6 - 3x_1 - 2x_2) + \frac{1}{4}(3x_1 - 2x_2) \geq \frac{1}{2}$$

$$\Downarrow$$

$$x_2 \leq 1$$

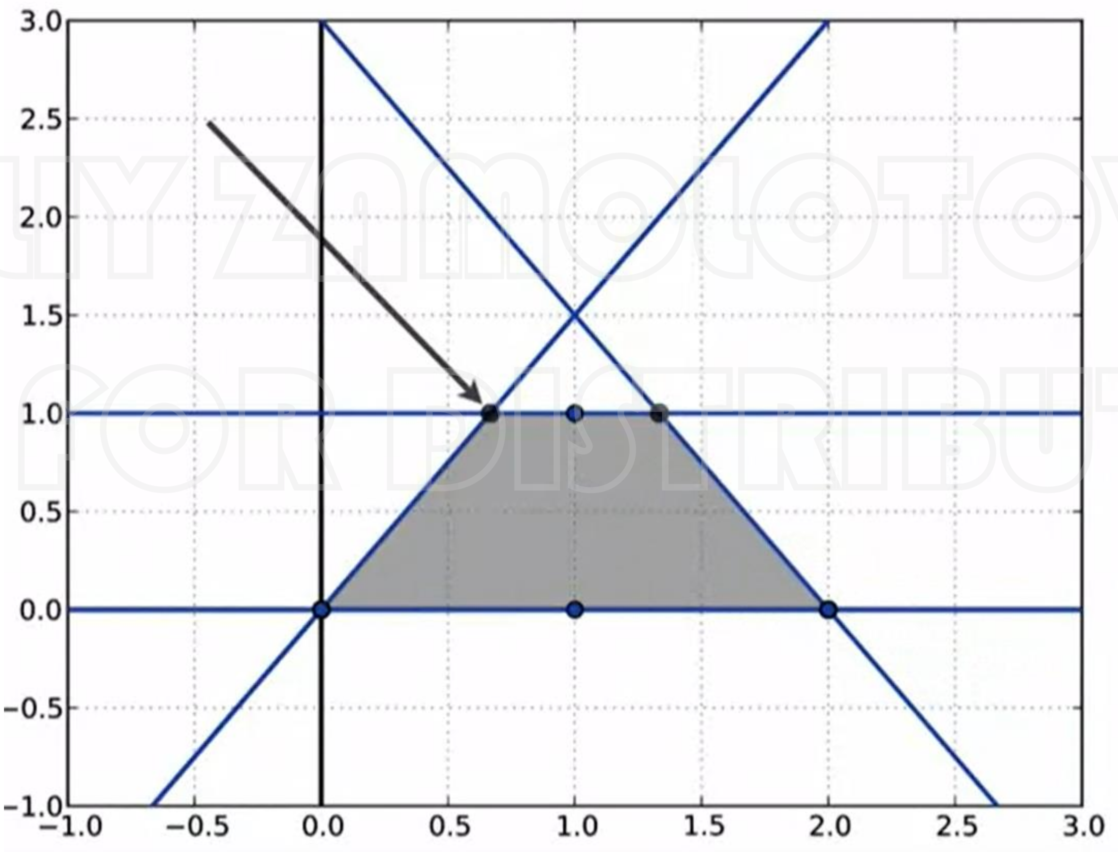
Сечение Гомори



Сечение Гомори

	x_1	x_2	x_3	x_4	s_1	b
	0	0	1/4	1/4	0	3/2
x_1	1	0	1/6	-1/6	0	1
x_2	0	1	1/4	1/4	0	3/2
s_1	0	0	-1/4	-1/4	1	-1/2

Сечение Гомори



Сечение Гомори

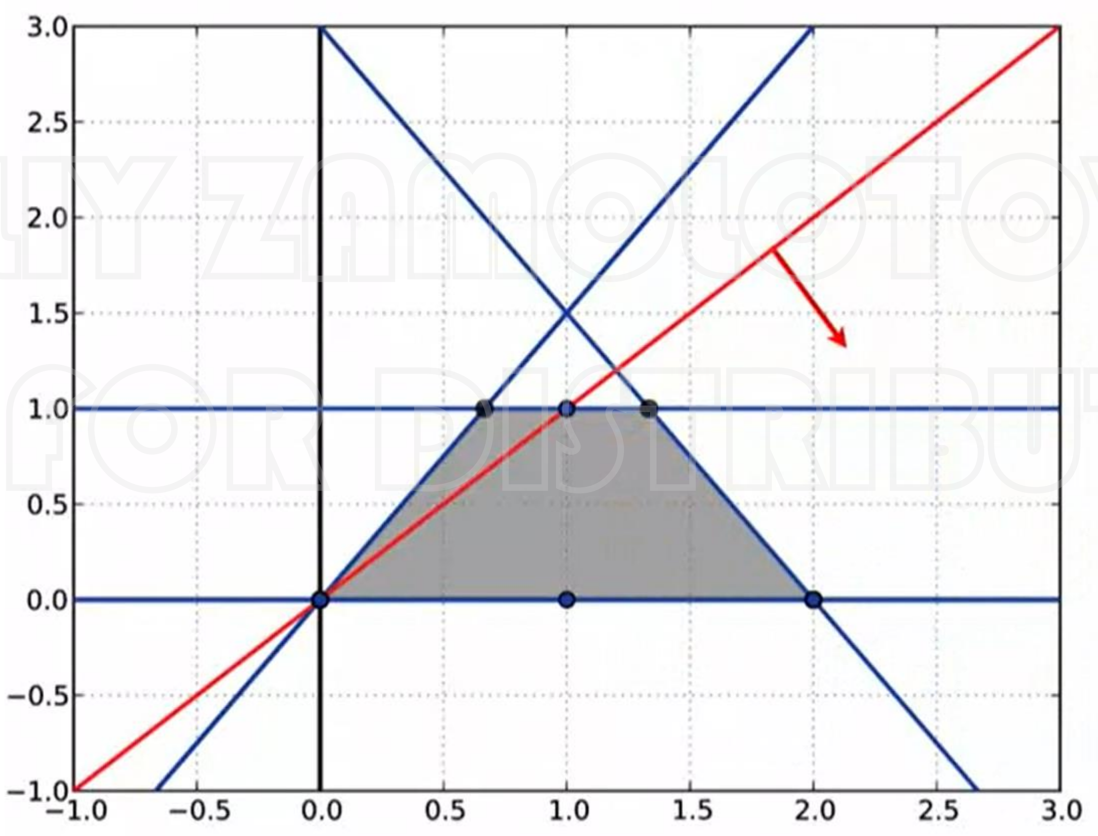
	x_1	x_2	x_3	x_4	s_1	b
	0	0	0	0	0	1
x_1	1	0	0	-1/3	2/3	2/3
x_2	0	1	0	0	1	1
s_1	0	0	1	1	-4	2

$$\frac{2}{3}x_4 + \frac{2}{3}s_1 \geq \frac{2}{3}$$

- Выразим через x_1 и x_2

$$x_1 - x_2 \geq 0$$

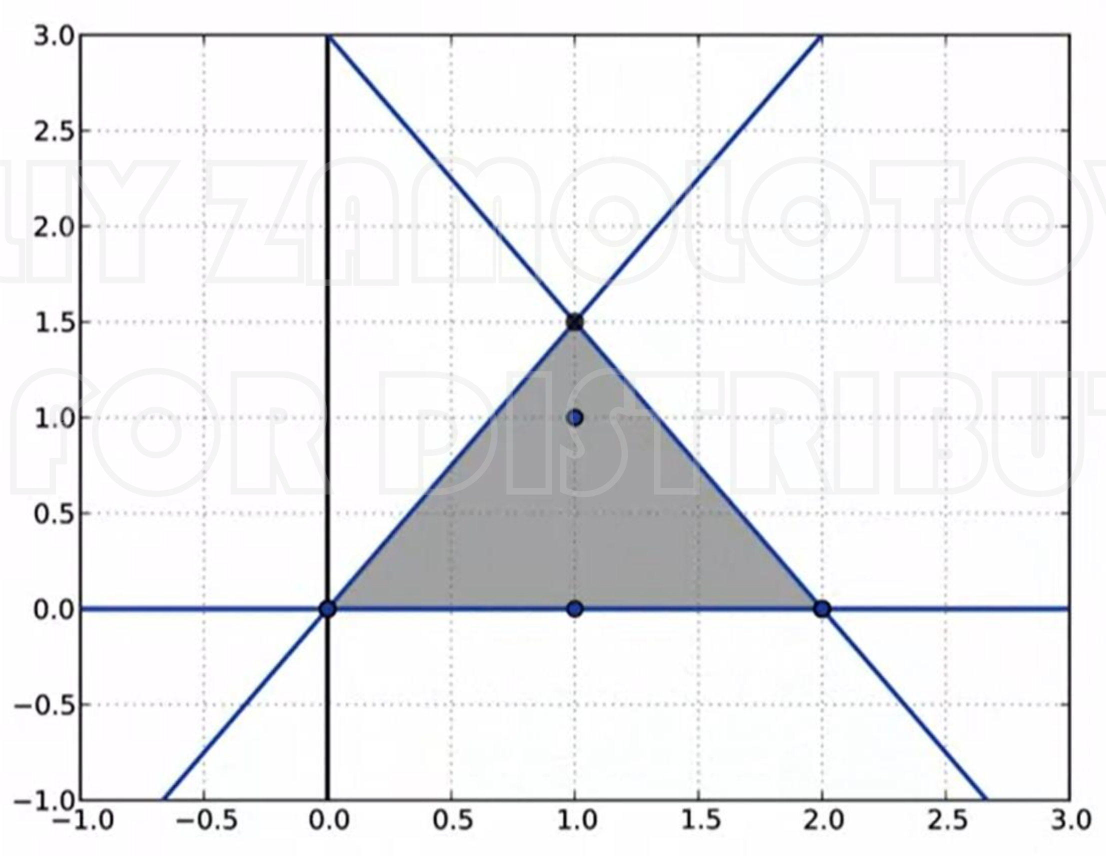
Сечение Гомори



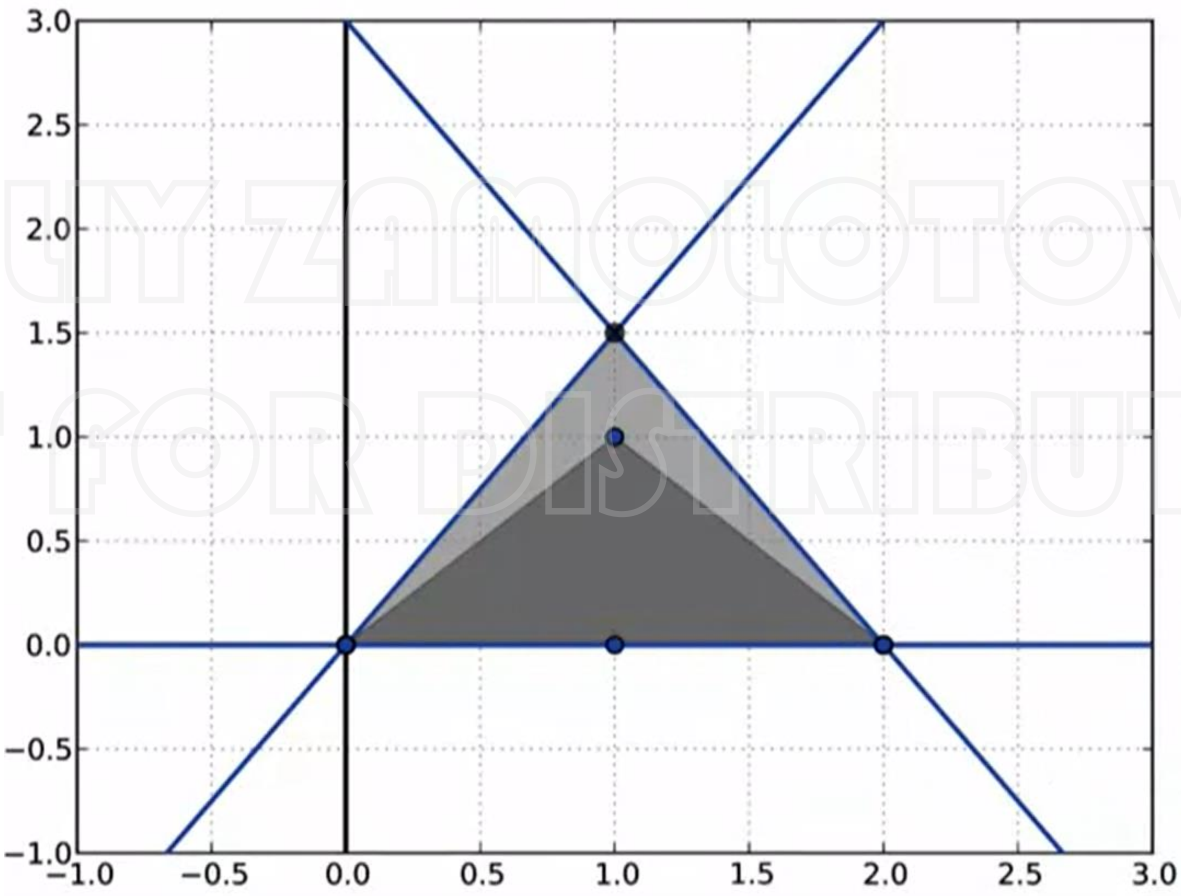
Смешанное целочисленное программирование

Выпуклая оболочка, сечение многогранником

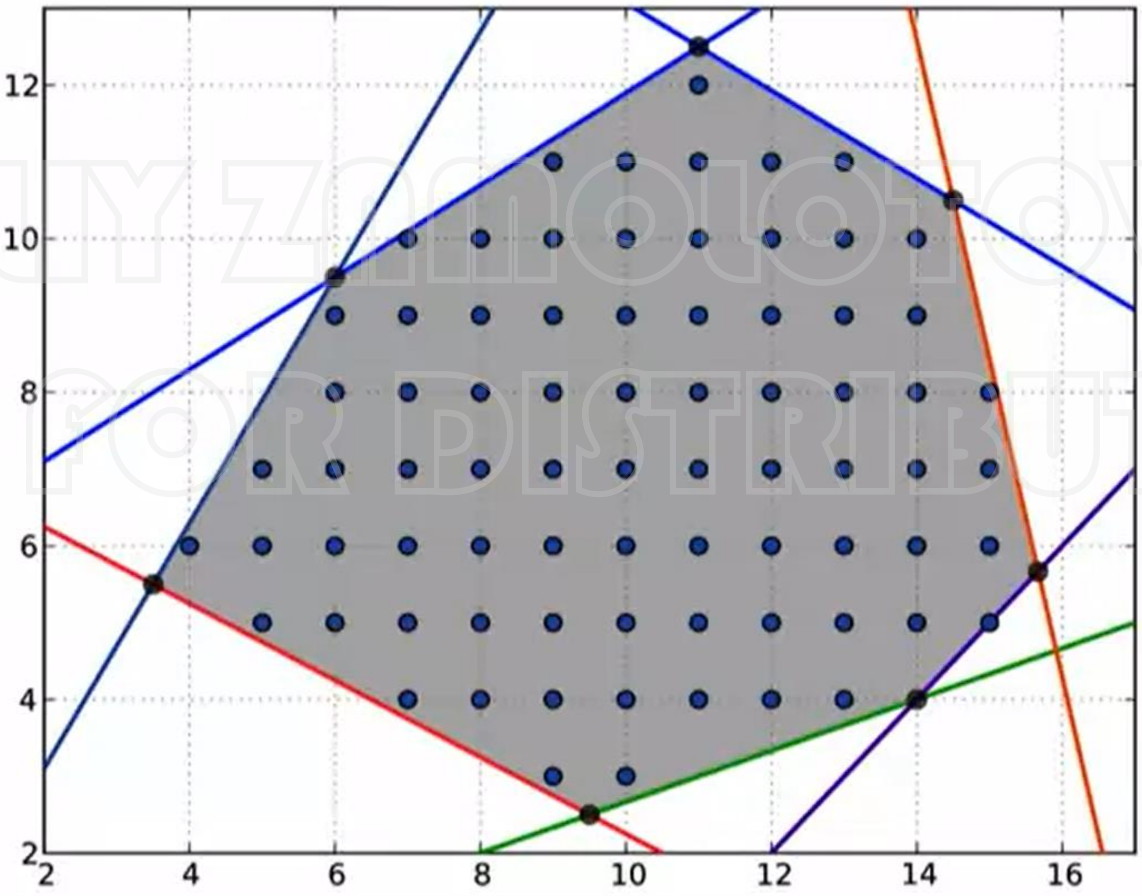
Выпуклая оболочка



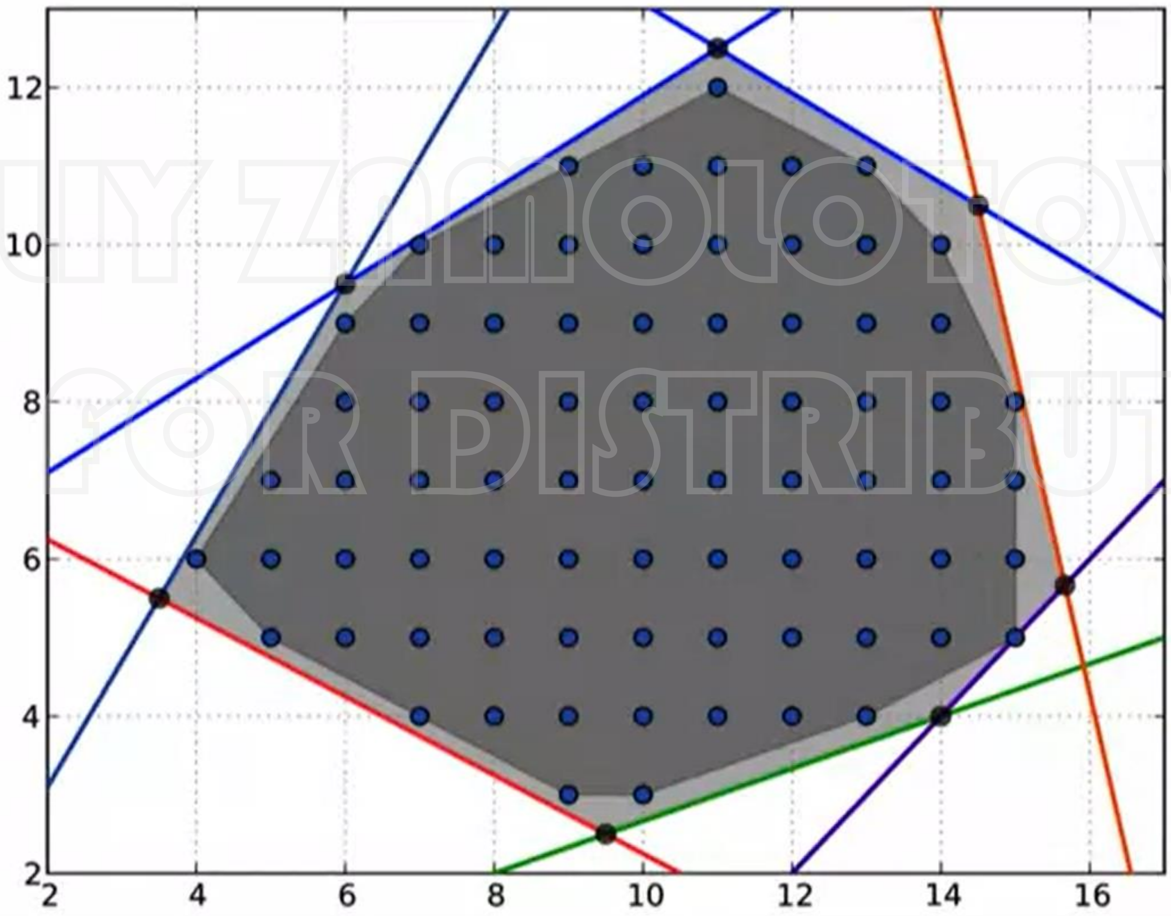
Выпуклая оболочка



Выпуклая оболочка



Выпуклая оболочка



Сечение многогранником

- Сечение многогранником
  - Отсекает области, образованные выпуклой оболочкой
- Сечение корректно:
  - Не убирает ни одного решения
- Сечение сильное, насколько это возможно
  - Если мы имеем все из них, то можно решить задачу линейного программирования

Сечение многогранником

- Использует структуру задачи
  - Строится на базе ограничений задачи
- Должен удалить текущее базовое выполнимое решение
- Не нужно генерировать все из них
- Приложение может использовать разные типы отсечений

Грани

- Для того, чтобы найти грань в R^n
  - Найти n аффинно независимых решений (точек), удовлетворяющих ограничению, как равенству
- Аффинная независимость
  - $x_1, .., x_n$ аффинно независимы тогда и только тогда, когда $(x_1, 1), \dots, (x_n, 1)$ – линейно независимы

Расположение складов

$$Minimize \sum_{w \in W} c_w x_w + \sum_{w \in W, c \in C} t_{w,c}\, y_{w,c}$$

При этом $y_{wc} \le x_w \; (w \in W, c \in C)$

$$\sum_{w \in W} y_{wc} = 1 \; (c \in C)$$

$$x_w \in \{0,1\} \; (w \in W)$$

$$y_{wc} \in \{0,1\} \; (w \in W, c \in C)$$

Грани

- Какие будут грани для неравенства
  - $y_{wc} \leq x_w$
- Рассмотрим $y_{w,1} \leq x_w$ и найдем n точек

w	1	2	..	n	
0	0	0	…	0	1
1	1	0	..	0	1
1	1	1	..	0	1
..	..	..	…	…	…
1	1	0	0	1	1

Смешанное целочисленное программирование

Сечение покрытием, метод ветвей и сечения

Сечение покрытием

- Рассмотрим ограничения типа
  - $\sum_{j=1..n} a_j x_j \leq b$
- Какие грани будут у этого ограничения?
- Покрытие
  - Множество $C \subseteq N = \{1, .., n\}$ является покрытием, если $\sum_{j \in C} a_j > b$
  - Покрытие минимально, если C\{j} не покрытие для любого $j \in$ C
- Если
  - Множество $C \subseteq N = \{1, .., n\}$ является покрытием, тогда $\sum_{j \in C} x_j \leq |C| - 1$ соответствует неравенству

Сечение покрытием

- Рассмотрим ограничения типа
  - $11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19$
- Несколько минимальных покрытий для этого неравенства
  - $x_1 + x_2 + x_3 \leq 2$
  - $x_3 + x_4 + x_5 + x_6 \leq 3$

Сильнее сечения покрытием

- Если
  - Множество $C \subseteq N = \{1,..,n\}$ является покрытием, тогда $\sum_{j \in E(C)} x_j \le |C| - 1$ соответствует неравенству $\sum_{j=1..n} a_j x_j \le b$, где $\mathrm{E}(C) = C \cup \{\, j \mid \forall i \in C\colon a_j \ge a_i\}$
- Рассмотрим ограничения типа
  - $11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \le 19$
- И
  - $x_3 + x_4 + x_5 + x_6 \le 3$
- Сильнее, чем покрытие неравенства
  - $x_1 + \cdots + x_6 \le 3$

Метод ветвей и сечения

- Основная идея
  - 1. Формулируем задачу как СЦП
  - 2. Решаем линейную релаксацию, если линейная релаксация целая, останавливаемся
  - 3. Ищем многогранник, который усекает линейную релаксацию, если это возможно. Если возможно, то идём в пункт 2
  - 4. В противном случае, идём на другую ветку

Отделение для сечения покрытием

- Покрытие неравенства $\sum_{j\in C} x_j \le |C| - 1$
- Может быть записано как
  - $\sum_{j\in C}(1 - x_j) \ge 1$
- Это означает, что существует $C \subseteq N$ удовлетворяющее
  - $\sum_{j\in C}(1 - x_j^*) < 1$
  - $\sum_{j\in C} a_j > b$
- Это эквивалентно задаче
  - $min \sum_{j\in N}(1 - x_j^*) z_j$
  - При этом $\sum_{j\in N} a_j z_j > b \;\; z_j \;\in \{0,1\}$
- Если минимум этой задачи ниже 1, значит есть сечение, все переменные имеют значение 1 для сечения.

Отделение для сечения покрытием

- Рассмотрим ограничения типа
  - $45x_1 + 46x_2 + 79x_3 + 54x_4 + 53x_5 + 125x_6 \leq 178$
- Имеет дробное решение
  - $x^* = (0,\ 0,\ \frac{3}{4},\ \frac{1}{2},\ 1,\ 0)$
- Задача отделения
  - $\min z_1 + z_2 + \frac{1}{4} z_3 + \frac{1}{2} z_4 + z_6$
  - При этом $45z_1 + 46z_2 + 79z_3 + 54z_4 + 53z_5 + 125z_6 > 178$