# Смешанно-целочисленное программирование

Виктор Васильевич Лепин

# Задача смешанно-целочисленного программирования

- Задача смешанно-целочисленного программирования (СЦП) есть следующая оптимизационная задача: $\max\{c^T x : b^1 \leq Ax \leq b^2, d^1 \leq x \leq d^2, \;\; x_j \in \mathbb{Z} \text{ для } j \in S\}$, где
  - $b^1$, $b^2 \in \mathbb{R}^m$,

# Задача смешанно-целочисленного программирования

- Задача смешанно-целочисленного программирования (СЦП) есть следующая оптимизационная задача: $\max\{c^T x : b^1 \leq Ax \leq b^2, d^1 \leq x \leq d^2, \ \ x_j \in \mathbb{Z} \text{ для } j \in S\}$, где
  - $b^1$, $b^2 \in \mathbb{R}^m$,
  - $c$, $d^1$, $d^2 \in \mathbb{R}^n$,

# Задача смешанно-целочисленного программирования

- Задача смешанно-целочисленного программирования (СЦП) есть следующая оптимизационная задача: $\max\{c^T x : b^1 \leq Ax \leq b^2, d^1 \leq x \leq d^2, \ \ x_j \in \mathbb{Z}$ для $j \in S\}$, где
  - $b^1$, $b^2 \in \mathbb{R}^m$,
  - $c$, $d^1$, $d^2 \in \mathbb{R}^n$,
  - $A$ — действительная $m \times n$-матрица,

# Задача смешанно-целочисленного программирования

- Задача смешанно-целочисленного программирования (СЦП) есть следующая оптимизационная задача: $\max\{c^T x : b^1 \le Ax \le b^2, d^1 \le x \le d^2, \;\; x_j \in \mathbb{Z} \text{ для } j \in S\}$, где
    - $b^1, b^2 \in \mathbb{R}^m$,
    - $c, d^1, d^2 \in \mathbb{R}^n$,
    - $A$ — действительная $m \times n$-матрица,
    - $x$ — $n$-вектор переменных (неизвестных),

# Задача смешанно-целочисленного программирования

- Задача смешанно-целочисленного программирования (СЦП) есть следующая оптимизационная задача: $\max\{c^T x : b^1 \le Ax \le b^2, d^1 \le x \le d^2, \;\; x_j \in \mathbb{Z} \text{ для } j \in S\}$, где
  - $b^1$, $b^2 \in \mathbb{R}^m$,
  - $c$, $d^1$, $d^2 \in \mathbb{R}^n$,
  - $A$ — действительная $m \times n$-матрица,
  - $x$ — $n$-вектор переменных (неизвестных),
  - а $S \subseteq \{1, \dots, n\}$ есть множество целочисленных переменных.

# Задача смешанно-целочисленного программирования

- Задача смешанно-целочисленного программирования (СЦП) есть следующая оптимизационная задача: $\max\{c^T x : b^1 \leq Ax \leq b^2, d^1 \leq x \leq d^2, \ \ x_j \in \mathbb{Z}$ для $j \in S\}$, где
  - $b^1$, $b^2 \in \mathbb{R}^m$,
  - $c$, $d^1$, $d^2 \in \mathbb{R}^n$,
  - $A$ — действительная $m \times n$-матрица,
  - $x$ — $n$-вектор переменных (неизвестных),
  - а $S \subseteq \{1, \ldots, n\}$ есть множество целочисленных переменных.
- В задаче целочисленного программирования (ЦП) все переменные целочисленны ($|S| = n$).

# Отличие задач СЦП и ЛП

- Задача СЦП отличается от задачи линейного программирования (ЛП) тем,

# Отличие задач СЦП и ЛП

- Задача СЦП отличается от задачи линейного программирования (ЛП) тем,
- что некоторые переменные могут принимать значения из дискретного множества.

# Отличие задач СЦП и ЛП

- Задача СЦП отличается от задачи линейного программирования (ЛП) тем,
- что некоторые переменные могут принимать значения из дискретного множества.
- Это отличие делает задачу СЦП
    - существенно более полезной на практике,

# Отличие задач СЦП и ЛП

- Задача СЦП отличается от задачи линейного программирования (ЛП) тем,
- что некоторые переменные могут принимать значения из дискретного множества.
- Это отличие делает задачу СЦП
  - существенно более полезной на практике,
  - но существенно сложнее с алгоритмической точки зрения.

# Отличие задач СЦП и ЛП

- Задача СЦП отличается от задачи линейного программирования (ЛП) тем,
- что некоторые переменные могут принимать значения из дискретного множества.
- Это отличие делает задачу СЦП
  - существенно более полезной на практике,
  - но существенно сложнее с алгоритмической точки зрения.
- Можно сказать, что задача СЦП — это одна из самых трудных задач математического программирования.

# Отличие задач СЦП и ЛП

- Задача СЦП отличается от задачи линейного программирования (ЛП) тем,
- что некоторые переменные могут принимать значения из дискретного множества.
- Это отличие делает задачу СЦП
  - существенно более полезной на практике,
  - но существенно сложнее с алгоритмической точки зрения.
- Можно сказать, что задача СЦП — это одна из самых трудных задач математического программирования.
- И это неудивительно, поскольку даже самые трудные комбинаторные задачи очень просто формулируются как задачи СЦП.

# Отличие задач СЦП и ЛП

- Задача СЦП отличается от задачи линейного программирования (ЛП) тем,
- что некоторые переменные могут принимать значения из дискретного множества.
- Это отличие делает задачу СЦП
  - существенно более полезной на практике,
  - но существенно сложнее с алгоритмической точки зрения.
- Можно сказать, что задача СЦП — это одна из самых трудных задач математического программирования.
- И это неудивительно, поскольку даже самые трудные комбинаторные задачи очень просто формулируются как задачи СЦП.
- Одно из самых распространенных применений СЦП в повседневной жизни касается эффективного использования ограниченных ресурсов.

# Фиксированные доплаты

- Одно из главных ограничений по применению линейного программирования для решения задач производственного планирования — это невозможность учесть фиксированные издержки.

# Фиксированные доплаты

- Одно из главных ограничений по применению линейного программирования для решения задач производственного планирования — это невозможность учесть фиксированные издержки.
- В моделях СЦП учет фиксированных издержек осуществляется просто.

Функция стоимости с фиксированными доплатами имеет вид

$$c(x) = \begin{cases} ax + b, & \text{если } 0 < l \le x \le u, \\ 0, & \text{если } x = 0, \end{cases}$$

где

- $a$ — переменные издержки,

# Фиксированные доплаты

Функция стоимости с фиксированными доплатами имеет вид

$$c(x) = \begin{cases} ax + b, & \text{если } 0 < l \le x \le u, \\ 0, & \text{если } x = 0, \end{cases}$$

где

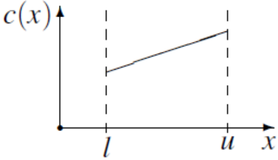
- $a$ — переменные издержки,
- $b$ — постоянные издержки.

Функция стоимости с фиксированными доплатами имеет вид

$$c(x) = \begin{cases} ax + b, & \text{если } 0 < l \le x \le u, \\ 0, & \text{если } x = 0, \end{cases}$$

где

- $a$ — переменные издержки,
- $b$ — постоянные издержки.




Ее график представлен на рис. справа.

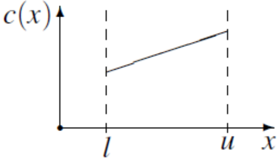
- Если ввести бинарную переменную $y$, которая принимает одно из двух значений 0 или 1,

Функция стоимости с фиксированными доплатами имеет вид

$$c(x) = \begin{cases} ax + b, & \text{если } 0 < l \leq x \leq u, \\ 0, & \text{если } x = 0, \end{cases}$$

где

- $a$ — переменные издержки,
- $b$ — постоянные издержки.




Ее график представлен на рис. справа.

- Если ввести бинарную переменную $y$, которая принимает одно из двух значений 0 или 1,
- и добавить неравенства $ly \leq x \leq uy$,

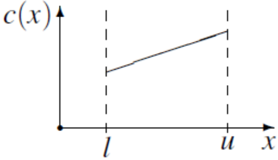
# Фиксированные доплаты

Функция стоимости с фиксированными доплатами имеет вид

$$c(x) = \begin{cases} ax + b, & \text{если } 0 < l \le x \le u, \\ 0, & \text{если } x = 0, \end{cases}$$

где

- $a$ — переменные издержки,
- $b$ — постоянные издержки.




Ее график представлен на рис. справа.

- Если ввести бинарную переменную $y$, которая принимает одно из двух значений 0 или 1,
- и добавить неравенства $ly \le x \le uy$,
- то функцию $c(x)$ можно преобразовать в линейную $c(x, y) = ax + by$.

- *Дискретная переменная* $x$ может принимать только конечное число значений $v_1, \ldots, v_k$.

# Дискретные переменные

- *Дискретная переменная* $x$ может принимать только конечное число значений $v_1, \dots, v_k$.
- Например, в задаче проектирования автомобиля объем двигателя $x$ может принимать, скажем, одно из четырех значений: 1.4, 1.6, 1.9 и 2.0 литра.

# Дискретные переменные

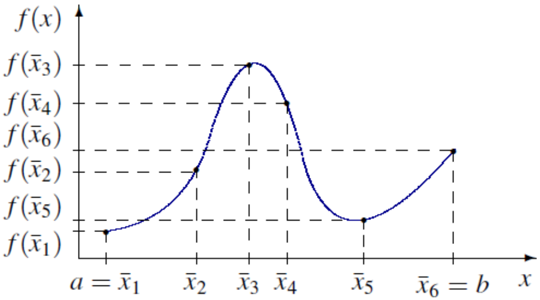
- *Дискретная переменная* $x$ может принимать только конечное число значений $v_1, \ldots, v_k$.
- Например, в задаче проектирования автомобиля объем двигателя $x$ может принимать, скажем, одно из четырех значений: 1.4, 1.6, 1.9 и 2.0 литра.
- Дискретную переменную $x$ можно представить как обычную непрерывную переменную,
    - вводя бинарные переменные $y_1, \ldots, y_k$

# Дискретные переменные

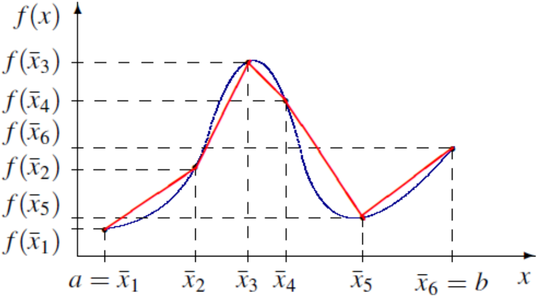
- *Дискретная переменная* $x$ может принимать только конечное число значений $v_1, \ldots, v_k$.
- Например, в задаче проектирования автомобиля объем двигателя $x$ может принимать, скажем, одно из четырех значений: 1.4, 1.6, 1.9 и 2.0 литра.
- Дискретную переменную $x$ можно представить как обычную непрерывную переменную,
  - вводя бинарные переменные $y_1, \ldots, y_k$
  - и записывая ограничения

$$
\begin{aligned}
&x - v_1 y_1 - v_2 y_2 - \cdots - v_k y_k = 0,\\
&y_1 + y_2 + \cdots + y_k = 1,\\
&y_i \in \mathbb{Z}_+, \quad i = 1, \ldots, k.
\end{aligned}
$$

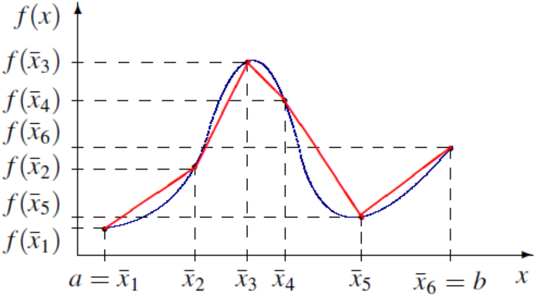
# Аппроксимация нелинейной функции




- Нелинейная функция $y = f(x)$ задана на отрезке $[a, b]$.
- Выберем разбиение $a = \bar{x}_1 < \bar{x}_2 < \cdots < \bar{x}_r = b$ отрезка $[a, b]$.
- Соединяя соседние точки $(\bar{x}_k, \bar{y}_k = f(\bar{x}_k))$ и $(\bar{x}_{k+1}, \bar{y}_{k+1} = f(\bar{x}_{k+1}))$ отрезками прямых,
- мы получим кусочно-линейную аппроксимацию $\tilde{f}(x)$ функции $f(x)$.




- Нелинейная функция $y = f(x)$ задана на отрезке $[a, b]$.
- Выберем разбиение $a = \bar{x}_1 < \bar{x}_2 < \cdots < \bar{x}_r = b$ отрезка $[a, b]$.
- Соединяя соседние точки $(\bar{x}_k, \bar{y}_k = f(\bar{x}_k))$ и $(\bar{x}_{k+1}, \bar{y}_{k+1} = f(\bar{x}_{k+1}))$ отрезками прямых,
- мы получим кусочно-линейную аппроксимацию $\tilde{f}(x)$ функции $f(x)$.




- Нелинейная функция $y = f(x)$ задана на отрезке $[a, b]$.
- Выберем разбиение $a = \bar{x}_1 < \bar{x}_2 < \cdots < \bar{x}_r = b$ отрезка $[a, b]$.
- Соединяя соседние точки $(\bar{x}_k, \bar{y}_k = f(\bar{x}_k))$ и $(\bar{x}_{k+1}, \bar{y}_{k+1} = f(\bar{x}_{k+1}))$ отрезками прямых,
- мы получим кусочно-линейную аппроксимацию $\tilde{f}(x)$ функции $f(x)$.

То, что точка $(x, y)^T$ лежит на графике функции $\tilde{f}$, выражается следующей системой ограничений:

# Аппроксимация нелинейной функции

То, что точка $(x, y)^T$ лежит на графике функции $\tilde{f}$, выражается следующей системой ограничений:
$0 \leq \lambda_k \leq \delta_k$, $\delta_k \in \{0, 1\}$, $k = 1, \ldots, r$,

То, что точка $(x, y)^T$ лежит на графике функции $\tilde{f}$, выражается следующей системой ограничений:
$0 \le \lambda_k \le \delta_k$, $\delta_k \in \{0, 1\}$, $k = 1, \ldots, r$,
$\delta_i + \delta_j \le 1$, $j = 3, \ldots, r$; $i = 1, \ldots, j - 2$,

То, что точка $(x, y)^T$ лежит на графике функции $\tilde{f}$, выражается следующей системой ограничений:
$0 \le \lambda_k \le \delta_k$, $\delta_k \in \{0, 1\}$, $k = 1, \ldots, r$,
$\delta_i + \delta_j \le 1$, $j = 3, \ldots, r$; $i = 1, \ldots, j-2$,

$$x = \sum_{k=1}^{r} \lambda_k \bar{x}_k, \qquad y = \sum_{k=1}^{r} \lambda_k \bar{y}_k, \qquad \sum_{k=1}^{r} \lambda_k = 1.$$

# Аппроксимация нелинейной функции

То, что точка $(x, y)^T$ лежит на графике функции $\tilde{f}$, выражается следующей системой ограничений:

$0 \le \lambda_k \le \delta_k$, $\delta_k \in \{0, 1\}$, $k = 1, \ldots, r$,

$\delta_i + \delta_j \le 1$, $j = 3, \ldots, r$; $i = 1, \ldots, j - 2$,

$$x = \sum_{k=1}^{r} \lambda_k \bar{x}_k, \qquad y = \sum_{k=1}^{r} \lambda_k \bar{y}_k, \qquad \sum_{k=1}^{r} \lambda_k = 1.$$

- Первые две группы ограничений означают, что среди чисел $\lambda_1, \ldots, \lambda_r$ не более двух ненулевых, причем, если два числа $\lambda_{j_1}$ и $\lambda_{j_2}$ ненулевые, то они соседние, т. е. $|j_1 - j_2| = 1$.

# Аппроксимация нелинейной функции

То, что точка $(x, y)^T$ лежит на графике функции $\tilde{f}$, выражается следующей системой ограничений:

$0 \le \lambda_k \le \delta_k$, $\delta_k \in \{0, 1\}$, $k = 1, \dots, r$,

$\delta_i + \delta_j \le 1$, $j = 3, \dots, r$; $i = 1, \dots, j - 2$,

$$x = \sum_{k=1}^{r} \lambda_k \bar{x}_k, \qquad y = \sum_{k=1}^{r} \lambda_k \bar{y}_k, \qquad \sum_{k=1}^{r} \lambda_k = 1.$$

- Первые две группы ограничений означают, что среди чисел $\lambda_1, \dots, \lambda_r$ не более двух ненулевых, причем, если два числа $\lambda_{j_1}$ и $\lambda_{j_2}$ ненулевые, то они соседние, т. е. $|j_1 - j_2| = 1$.
- Следующие три равенства означают, что точка $(x, y)^T$ принадлежит выпуклой оболочке точек $(\bar{x}_k, f(\bar{x}_k))^T$, $k = 1, \dots, r$.

То, что точка $(x, y)^T$ лежит на графике функции $\tilde{f}$, выражается следующей системой ограничений:

$0 \leq \lambda_k \leq \delta_k$, $\delta_k \in \{0, 1\}$, $k = 1, \dots, r$,

$\delta_i + \delta_j \leq 1$, $j = 3, \dots, r$; $i = 1, \dots, j-2$,

$$x = \sum_{k=1}^{r} \lambda_k \bar{x}_k, \qquad y = \sum_{k=1}^{r} \lambda_k \bar{y}_k, \qquad \sum_{k=1}^{r} \lambda_k = 1.$$

- Первые две группы ограничений означают, что среди чисел $\lambda_1, \dots, \lambda_r$ не более двух ненулевых, причем, если два числа $\lambda_{j_1}$ и $\lambda_{j_2}$ ненулевые, то они соседние, т. е. $|j_1 - j_2| = 1$.
- Следующие три равенства означают, что точка $(x, y)^T$ принадлежит выпуклой оболочке точек $(\bar{x}_k, f(\bar{x}_k))^T$, $k = 1, \dots, r$.
- Все условия вместе гарантируют, что точка $(x, y)^T$ лежит на графике функции $\tilde{f}(x)$.

## Булевы переменные и формулы

- Формально мы записываем логические условия с помощью булевых переменных и формул.

# Булевы переменные и формулы

- Формально мы записываем логические условия с помощью булевых переменных и формул.
- *Булева переменная* может принимать только два значения: **истина** и **ложь**.

# Булевы переменные и формулы

- Формально мы записываем логические условия с помощью булевых переменных и формул.
- *Булева переменная* может принимать только два значения: **истина** и **ложь**.
- Из булевых переменных с помощью бинарных логических операций $\vee$ (или), $\wedge$ (и)

# Булевы переменные и формулы

- Формально мы записываем логические условия с помощью булевых переменных и формул.
- *Булева переменная* может принимать только два значения: **истина** и **ложь**.
- Из булевых переменных с помощью бинарных логических операций $\vee$ (или), $\wedge$ (и)
- и унарной операции $\neg$ ($\neg x$ означает не $x$)

# Булевы переменные и формулы

- Формально мы записываем логические условия с помощью булевых переменных и формул.
- *Булева переменная* может принимать только два значения: **истина** и **ложь**.
- Из булевых переменных с помощью бинарных логических операций $\vee$ (или), $\wedge$ (и)
- и унарной операции $\neg$ ($\neg x$ означает не $x$)
- можно образовывать *булевы формулы* почти так же, как из действительных переменных с помощью арифметических операций можно образовывать алгебраические выражения.

# Булевы переменные и формулы

- Формально мы записываем логические условия с помощью булевых переменных и формул.
- *Булева переменная* может принимать только два значения: **истина** и **ложь**.
- Из булевых переменных с помощью бинарных логических операций $\vee$ (или), $\wedge$ (и)
- и унарной операции $\neg$ ($\neg x$ означает не $x$)
- можно образовывать *булевы формулы* почти так же, как из действительных переменных с помощью арифметических операций можно образовывать алгебраические выражения.
- Например, $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$ есть булева формула.

# Таблицы истинности логических операций

- Логическая операция $\neg$

| $x$ | $\neg x$ |
|---|---|
| **ложь** | **истина** |
| **истина** | **ложь** |

- Логические операции $\wedge$ и $\vee$

| $x_1$ | $x_2$ | $x_1 \wedge x_2$ | $x_1 \vee x_2$ |
|---|---|---|---|
| **ложь** | **ложь** | **ложь** | **ложь** |
| **истина** | **ложь** | **ложь** | **истина** |
| **ложь** | **истина** | **ложь** | **истина** |
| **истина** | **истина** | **истина** | **истина** |

- Подставляя значения для булевых переменных, мы можем вычислить значение булевой формулы.

# Вычисление значения булевой формулы

- Подставляя значения для булевых переменных, мы можем вычислить значение булевой формулы.
- Например, для набора истинности $(x_1, x_2, x_3)$ =(**истина,ложь,ложь**)

# Вычисление значения булевой формулы

- Подставляя значения для булевых переменных, мы можем вычислить значение булевой формулы.
- Например, для набора истинности $(x_1, x_2, x_3) =$(**истина,ложь,ложь**)
- булева формула $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$

# Вычисление значения булевой формулы

- Подставляя значения для булевых переменных, мы можем вычислить значение булевой формулы.
- Например, для набора истинности $(x_1, x_2, x_3) =$(**истина,ложь,ложь**)
- булева формула $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$
- принимает значение (**истина**$\vee\neg$**ложь**)$\wedge$($\neg$**истина**$\vee$**ложь**) $=$ **истина**$\wedge$**ложь** $=$ **ложь.**

- Любую булеву формулу n булевых переменных можно представить в виде конъюнктивной нормальной формы (КНФ):

$$\bigwedge_{i=1}^{m} \left( \bigvee_{j \in S_i} x_j^{\sigma_j^i} \right),$$

# Конъюнктивная нормальная форма (КНФ)

- Любую булеву формулу n булевых переменных можно представить в виде конъюнктивной нормальной формы (КНФ):
$$\bigwedge_{i=1}^{m}\left(\bigvee_{j\in S_i} x_j^{\sigma_j^i}\right),$$
- где $S_i \subseteq \{1,\ldots,n\}$ $(i=1,\ldots,m)$ и все $\sigma_i \in \{0,1\}$.

- Любую булеву формулу n булевых переменных можно представить в виде конъюнктивной нормальной формы (КНФ):

$$\bigwedge_{i=1}^{m} \left( \bigvee_{j \in S_i} x_j^{\sigma_j^i} \right),$$

- где $S_i \subseteq \{1, \ldots, n\}$ $(i = 1, \ldots, m)$ и все $\sigma_i \in \{0, 1\}$.
- Здесь мы использовали следующие обозначения: $x^1 \stackrel{\mathsf{def}}{=} x$ и $x^0 \stackrel{\mathsf{def}}{=} \neg x$.

# Наборы истинности КНФ

- КНФ $\bigwedge_{i=1}^{m}\left(\bigvee_{j\in S_i} x_j^{\sigma_j^i}\right)$,

# Наборы истинности КНФ

- КНФ $\bigwedge_{i=1}^{m}\left(\bigvee_{j\in S_i} x_j^{\sigma_j^i}\right)$,
- принимает значение **истина** тогда и только тогда, когда

# Наборы истинности КНФ

- КНФ $\bigwedge_{i=1}^{m} \left( \bigvee_{j \in S_i} x_j^{\sigma_j^i} \right)$,
- принимает значение **истина** тогда и только тогда, когда
- каждый ее дизъюнкт $\left( \bigvee_{j \in S_i} x_j^{\sigma_j^i} \right)$ содержит хотя бы один литерал (литералом называется переменная или ее отрицание) со значением **истина**.

# Наборы истинности КНФ

- КНФ $\bigwedge_{i=1}^{m}\left(\bigvee_{j\in S_i} x_j^{\sigma_j^i}\right)$,
- принимает значение **истина** тогда и только тогда, когда
- каждый ее дизъюнкт $\left(\bigvee_{j\in S_i} x_j^{\sigma_j^i}\right)$ содержит хотя бы один литерал (литералом называется переменная или ее отрицание) со значением **истина**.
- Если отождествить значения **ложь** с 0, а **истина** с 1, то операция отрицания $\neg$ превращает $x$ в $1-x$.

# Наборы истинности КНФ

- КНФ $\bigwedge_{i=1}^{m} \left( \bigvee_{j \in S_i} x_j^{\sigma_j^i} \right)$,
- принимает значение **истина** тогда и только тогда, когда
- каждый ее дизъюнкт $\left( \bigvee_{j \in S_i} x_j^{\sigma_j^i} \right)$ содержит хотя бы один литерал (литералом называется переменная или ее отрицание) со значением **истина**.
- Если отождествить значения **ложь** с 0, а **истина** с 1, то операция отрицания $\neg$ превращает $x$ в $1 - x$.
- С учетом сказанного наборы истинности, на которых КНФ $\bigwedge_{i=1}^{m} \left( \bigvee_{j \in S_i} x_j^{\sigma_j^i} \right)$ принимает значение **истина**, являются решениями следующей системы неравенств:

$$\sum_{j \in S_i : \sigma_j^i = 1} x_j + \sum_{j \in S_i : \sigma_j^i = 0} (1 - x_j) \geq 1, \; i = 1, \ldots, m,$$

$$x_j \in \{0, 1\}, \; j = 1, \ldots, n,$$

# Наборы истинности КНФ: пример

- КНФ
  $(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_1)$
- принимает значение **истина** на наборах, которые являются решениями системы
  $x_1 + x_2 + x_3 \geq 1,$
  $x_1 + (1 - x_2) \geq 1,$
  $x_2 + (1 - x_3) \geq 1,$
  $x_3 + (1 - x_1) \geq 1,$
  $x_1, x_2, x_3 \in \{0, 1\}.$

# Задача бинарного программирования

- Требуется, чтобы из $m$ неравенств
  $A_i x \leq b_i, \;\; i = 1, \ldots, m,$
  выполнялись не менее $q$ любых неравенств.

# Задача бинарного программирования

- Требуется, чтобы из $m$ неравенств
  $A_i x \leq b_i, \;\; i = 1, \ldots, m,$
  выполнялись не менее $q$ любых неравенств.
- Например, если два задания $i$ и $j$ должны выполняться на одной машине, то мы должны потребовать выполнения следующей дизъюнкции:
  $e_i - s_j \leq 0$ или $e_j - s_i \leq 0$,

- Требуется, чтобы из $m$ неравенств
  $A_i x \leq b_i, \;\; i = 1, \ldots, m,$
  выполнялись не менее $q$ любых неравенств.
- Например, если два задания $i$ и $j$ должны выполняться на одной машине, то мы должны потребовать выполнения следующей дизъюнкции:
  $e_i - s_j \leq 0$ или $e_j - s_i \leq 0$,
- где $s_i$ и $e_i$ есть соответственно время начала и завершения задания $i$.

# Задача бинарного программирования

- Требуется, чтобы из $m$ неравенств
  $A_i x \leq b_i, \;\; i = 1, \ldots, m,$
  выполнялись не менее $q$ любых неравенств.

- Требуется, чтобы из $m$ неравенств
  $A_i x \leq b_i, \;\; i = 1, \ldots, m,$
  выполнялись не менее $q$ любых неравенств.
- Пусть $M$ есть достаточно большое число, такое, что неравенства $A_i x \leq b_i + M$ выполняются автоматически для всех допустимых векторов $x$ решаемой задачи.

- Требуется, чтобы из $m$ неравенств
  $A_i x \leq b_i, \;\; i = 1, \ldots, m,$
  выполнялись не менее $q$ любых неравенств.
- Пусть $M$ есть достаточно большое число, такое, что неравенства $A_i x \leq b_i + M$ выполняются автоматически для всех допустимых векторов $x$ решаемой задачи.
- Вводя бинарные переменные
  $$y_i = \begin{cases} 1, & \text{если ограничение } \; A_i x \leq b_i \text{ выполняется}, \\ 0, & \text{в противном случае}, \end{cases}$$

- Требуется, чтобы из $m$ неравенств
  $A_i x \le b_i, \ \ i = 1, \ldots, m,$
  выполнялись не менее $q$ любых неравенств.
- Пусть $M$ есть достаточно большое число, такое, что неравенства $A_i x \le b_i + M$ выполняются автоматически для всех допустимых векторов $x$ решаемой задачи.
- Вводя бинарные переменные
  $$y_i = \begin{cases} 1, & \text{если ограничение } \ A_i x \le b_i \text{ выполняется}, \\ 0, & \text{в противном случае}, \end{cases}$$
- мы можем учесть требуемое условие след. образом:
  $A_i x \le b_i + M(1 - y_i), \, i = 1, \ldots, m,$
  $$\sum_{i=1}^{m} y_i \ge q, \, y_i \in \{0, 1\}, \, i = 1, \ldots, k.$$