Смешанно-целочисленное программирование

Задача *смешанно-целочисленного программирования* (СЦП) есть следующая оптимизационная задача:

$$\max\{c^T x : b^1 \le Ax \le b^2,\ d^1 \le x \le d^2,\ x_j \in \mathbb{Z} \text{ для } j \in S\}, \qquad (5.1)$$

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. В задаче *целочисленного программирования* (ЦП) все переменные целочисленны ($|S| = n$).

Задача СЦП отличается от задачи *линейного программирования* (ЛП) тем, что некоторые переменные могут принимать значения из дискретного множества. Это отличие делает задачу СЦП существенно сложнее с алгоритмической точки зрения. Можно сказать, что задача СЦП — это одна из самых трудных задач математического программирования. И это неудивительно, поскольку многие комбинаторные задачи, включая те, которые считаются самыми трудными, очень просто формулируются как задачи СЦП. Одно из самых распространенных применений СЦП в повседневной жизни касается эффективного использования ограниченных ресурсов.

Целочисленность и нелинейность

Через условие «x — целое» можно выразить многие нелинейные ограничения. Но сначала мы покажем, что само это ограничение можно записать в непрерывных переменных при гладких ограничениях.

Условие, что x есть *бинарная переменная* (принимает только два значения: 0 и 1), записывается одним квадратичным равенством

$$x^2 - x = 0.$$

Такое представление бинарных переменных позволяет записывать многие задачи комбинаторной оптимизации как задачи квадратичного программирования. Для примера, трудная комбинаторная *задача о разбиении множества*

$$\max\{c^T x : Ax = \mathrm{e},\ x \in \{0,1\}^n\},$$

где $c \in \mathbb{R}^n$, а A есть $m \times n$-матрица с элементами 0 и 1, переписывается как задача квадратичного программирования следующим образом:

$$\begin{aligned} c^T x &\to \max, \\ Ax &= \mathrm{e}, \\ x_i^2 &= x_i\,, \quad i = 1, \ldots, n. \end{aligned}$$

Здесь и далее e обозначает вектор подходящего размера, все компоненты которого равны 1.

Предположим теперь, что целочисленная переменная x неотрицательна и ограничена сверху, т. е. $0 \le x \le d$, где d — положительное целое. Для записи числа d в двоичной системе счисления требуется $k = \lfloor \log d \rfloor + 1$ позиций. Поэтому мы можем представить условие $x \in \{0, 1, \ldots, d\}$ следующей системой уравнений:

$$x = \sum_{i=0}^{k-1} 2^i s_i \,,$$

$$s_i^2 = s_i, \quad i = 0, \ldots, k-1.$$

Итак, мы можем заключить, что задача СЦП сводится к задаче квадратичного программирования и, следовательно, не труднее последней. Но отличительная особенность целочисленного программирования состоит в том, что здесь целочисленность переменных учитывается совершенно особым образом на алгоритмическом уровне посредством ветвления по целочисленным переменным и генерации отсечений.

С практической точки зрения важным является, то что многие нелинейности моделируются введением целочисленных переменных

Фиксированные доплаты

Функция стоимости с фиксированными доплатами имеет вид

$$f(x) = \begin{cases} 0, & \text{если} \quad x = 0, \\ c_0 + cx, & \text{если} \quad x > 0, \end{cases}$$

где c_0 – некоторая фиксированная величина. Например, стоимость перевозки товара может состоять из двух частей: первая фиксирована и не зависит от объема перевозимого товара, а вторая – зависит. Предположим, что на переменные наложено требование неотрицательности $x \geq 0$ и известна верхняя оценка $x \leq \widehat{x}$ на величину неизвестных. Тогда путем введения новой целочисленной переменной такую задачу можно свести к ЗЦЛП.

Действительно, пусть переменная y удовлетворяет условиям:

$$y \in \{0, 1\},$$

$$y = 1 \quad \Leftrightarrow \quad x > 0.$$

Получаем

$$f(x) = c_0 y + cx.$$

$$y \in \{0, 1\},$$

$$x \le y\widehat{x}.$$

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

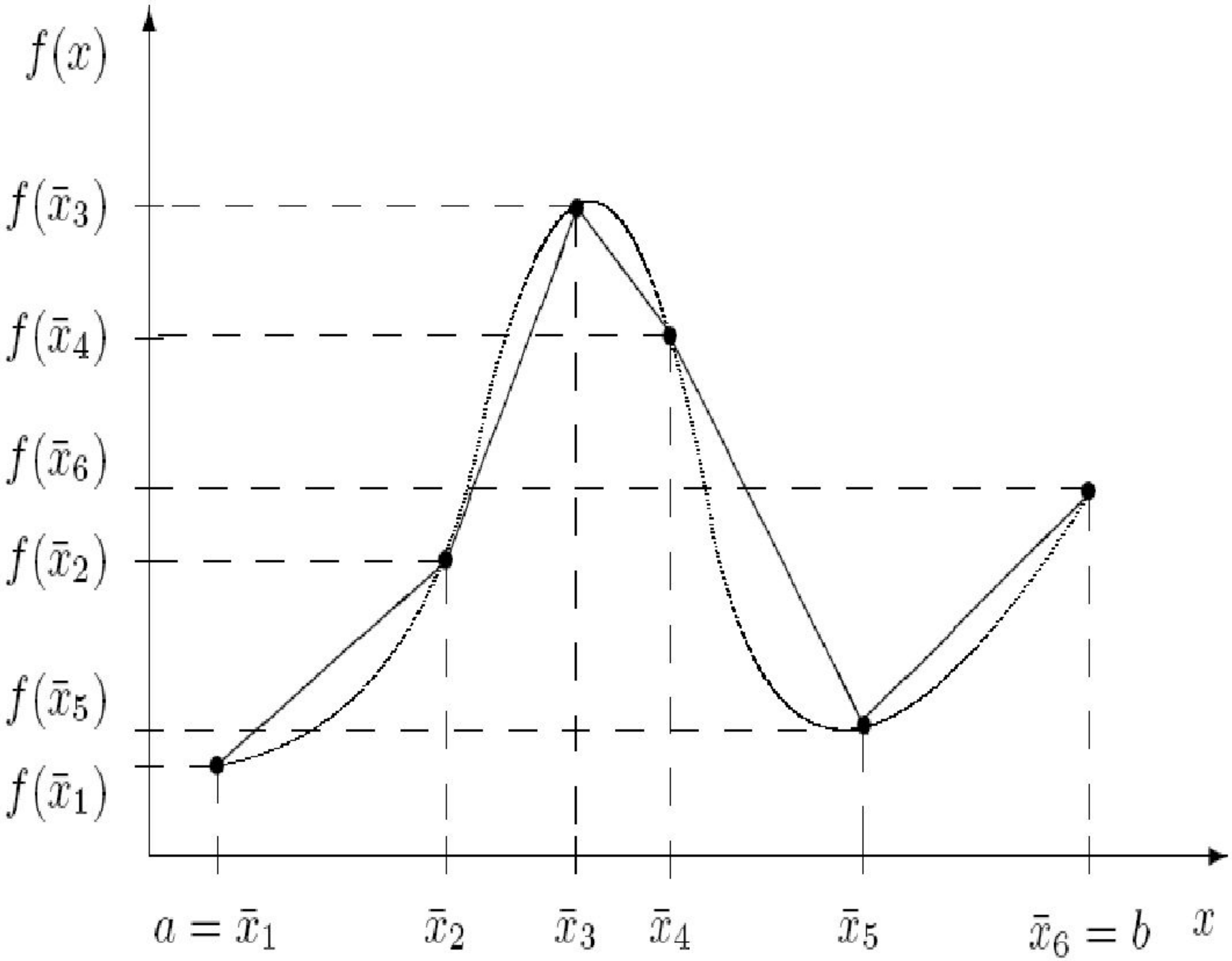
$$x - v_1 y_1 - v_2 y_2 - \ldots - v_k y_k = 0,$$

$$y_1 + y_2 + \ldots + y_k = 1,$$

$$y_i \in \mathbb{Z}_+, \quad i = 1, \ldots, k.$$

Не более одной переменной принимает ненулевое значение

Аппроксимация нелинейной функции



Кусочно-линейная аппроксимация нелинейной функции

Пусть нелинейная функция $y = f(x)$ задана на отрезке $[a, b]$. Выберем некоторое разбиение

$$a = \bar{x}_1 < \bar{x}_2 < \cdots < \bar{x}_r = b$$

отрезка $[a, b]$. Соединяя 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 $\tilde{f}(x)$ которая представляется следующей системой

$$x = \sum_{k=1}^{r} \lambda_k \bar{x}_k, \qquad \lambda_k \le \delta_k, \quad k = 1, \ldots, r,$$

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

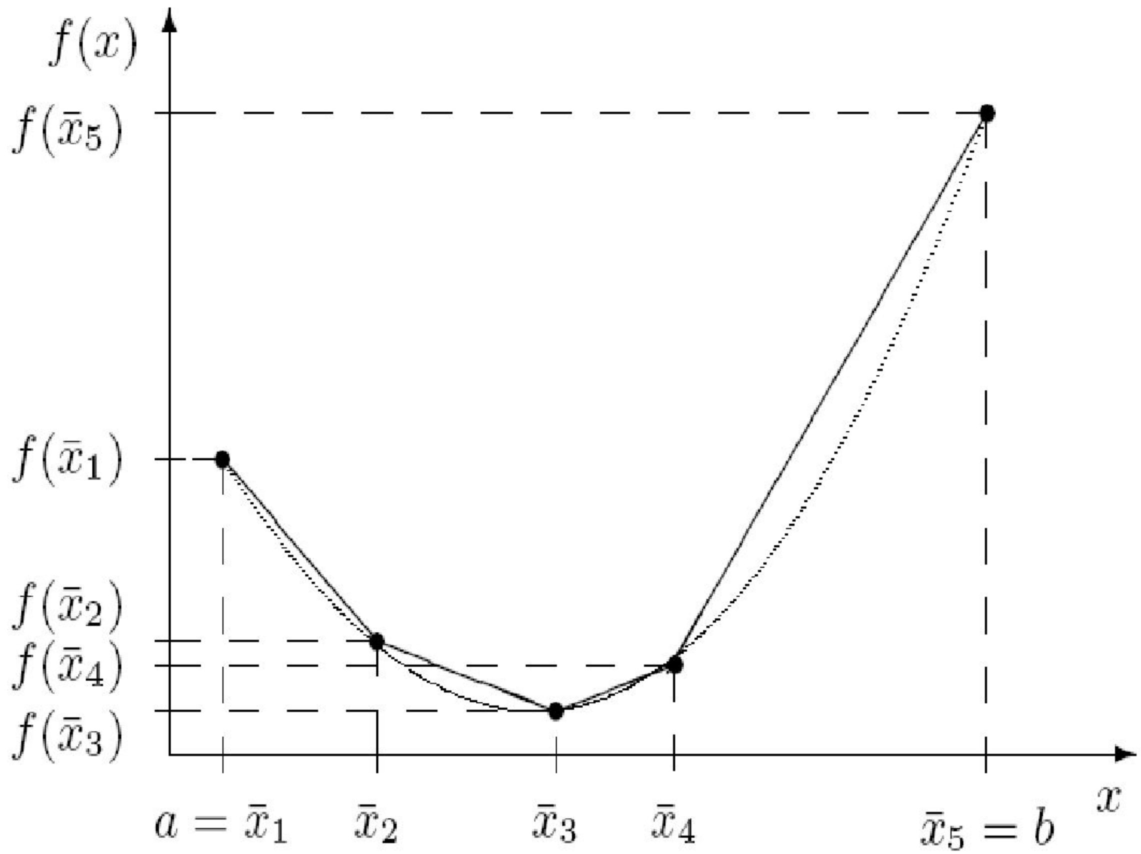
$$y = \sum_{k=1}^{r} \lambda_k \bar{y}_k, \qquad \lambda_k \ge 0,\ \delta_k \in \{0, 1\}, \quad k = 1, \ldots, r.$$

не более двух переменных λ_k принимают ненулевые значения

$$1 = \sum_{k=1}^{r} \lambda_k,$$

Аппроксимация выпуклой функции

Если функция $f(x)$ выпуклая, то во многих случаях мы можем представить зависимость $y = f(x)$ без введения целочисленных переменных. Как и ранее, аппроксимируем функцию f на интервале определения $[a, b]$ кусочно-линейной функцией $\tilde{f}$ с точками перегиба $a = \bar{x}_1 < \bar{x}_2 < \cdots < \bar{x}_r = b$



Для $k = 1, \dots, r-1$ определим числа

$$d_k = \bar{x}_{k+1} - \bar{x}_k,$$

$$q_k = \frac{f(\bar{x}_{k+1}) - f(\bar{x}_k)}{d_k}.$$

В силу выпуклости функции f имеем $q_1 \le q_2 \le \dots \le q_{r-1}$. Введем дополнительные действительные переменные x_k $(k = 1, \dots, r-1)$ и представим

$$x = \sum_{k=1}^{r-1} x_k,$$

$$y = f(a) + \sum_{k=1}^{r-1} q_k x_k,$$

$$0 \le x_k \le d_k, \quad k = 1, \dots, r-1.$$

Логические условия

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,

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$$

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

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

$$\sum_{j\in S_i^1} x_j + \sum_{j\in S_i^0} (1-x_j) \ge 1, \quad i=1,\ldots,m,$$

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

Например, 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

$$\begin{aligned} x_1 + x_2 + x_3 &\ge 1, \\ x_1 + (1-x_2) &\ge 1, \\ x_2 + (1-x_3) &\ge 1, \\ x_3 + (1-x_1) &\ge 1, \\ x_1,\ x_2,\ x_3 &\in \{0,1\}. \end{aligned}$$

Множественные альтернативы и дизъюнкции

Требуется, чтобы из m неравенств

$$A_i x \le b_i\,, \quad i = 1, \ldots, m,$$

выполнялись не менее q любых неравенств (не важно каких). Например, если два задания i и j должны выполняться на одной машине, то мы должны потребовать выполнения следующей дизъюнкции:

$$e_i - s_j \le 0 \quad \text{или} \quad e_j - s_i \le 0,$$

где s_i и e_i есть соответственно время начала и завершения задания i.

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 следующим образом:

$$A_i x \le b_i + M(1 - y_i), \quad i = 1, \ldots, m,$$

$$\sum_{i=1}^{m} y_i \ge q,$$

$$y_i \in \{0, 1\}, \quad i = 1, \ldots, k.$$

Здесь M — достаточно большое число, такое, что неравенства $A_i x \le b_i + M$ выполняются автоматически для всех допустимых векторов x решаемой задачи.

В заключение рассмотрим случай, когда из двух условий должно выполняться хотя бы одно:

$$x_1 \geq a \quad \text{или} \quad x_2 \geq b.$$

Например, мы хотим иметь рабочую станцию с $x_1 \geq a$ процессорами или однопроцессорную систему с частотой процессора $x_2 \geq b$.

Если обе переменные x_1 и x_2 неотрицательны, то, вводя бинарную переменную y, требуемую дизъюнкцию можно записать в виде

$$x_1 \geq \ ay, \quad x_2 \geq b(1-y).$$

Далее мы продемонстрируем использование множественных альтернатив и дизъюнкций на трех примерах.