Преподаватель: Власов Павел Александрович (ищи его в 1025л)

E-mail: pvlx@mail.ru

Лабораторныеработы: Пн, 19:10

Лекция 1

7 сентября 2015

1 Задачи методов оптимизации

1.1 Классификация задач оптимизации

Задача оптимизации имеет обычно имеет следующий вид:

$$\begin{cases} f(x) \to extr \\ x \in G \end{cases},$$

здесь f называется целевой функцией или критерием оптимальности, G называется множеством допустимых решений.

Замечание:

1. Задача min-целевой функции одной переменной:
$$\begin{cases} f(x) \to min \\ x \in [a, b] \subseteq \mathbb{R} \end{cases}$$
2. Задача безусловной оптимизации функции нескольких переменных:
$$\begin{cases} f(x) \to min \\ x \in G \subseteq \mathbb{R}^n \end{cases}$$
3. Задача условной оптимизации функции нескольких переменных:
$$\begin{cases} f(x) \to min \\ \phi(x) = 0 \\ x \in G \subseteq \mathbb{R}^n \end{cases}$$

1. Если $f : G \to \mathbb{R}$ - скалярная функция, то соотвующая задача называется задачей математическим программированием
2. Если $f : G \to \mathbb{R}^m, m \geq 2$, то соотвующая задача называется задачей многокретириальной оптимизации

Для задач математического программирования программирования дальнейшая классификация:

Вид функции	Стр. мно-ва G	Название задачи
Линейная	Выпуклый многоугольник в $\mathbb{R}^n$	Задача линейного программирования
Квадратичная	Выпуклый многоугольник в $\mathbb{R}^n$	Задача квадратичного программирования
Выпуклая	Выпуклое множеств в $\mathbb{R}^n$	Задача выпуклого программирования
Произвольная	Конечное множество	Задача дискретного программирования
Произвольная	Подмножество множества $\mathbb{Z}^n$	Задача целочисленного программирования
Произвольная	Подмножество в $\{0, 1\}^n$	Задача логического программирования

Замечание:

1. В этом семестре будем заниматься задачами линейного и целочисленного программирования. Задачами выпуклого и квадратичного программирования будем заниматься в следующем семестре.

2. Как правило задачи оптимизации, в которых мн-во G конечно или счетно, относят к разделу методов оптимизации, которые называется *исследованием операций.*

1.2 Венгерский метод решения задач о назначениях

1.2.1 Постановка задачи о назначениях

Содержательная постановка.

В распоряжении работодателя имеется n работ и n исполнителей. Стоимость выполнения i-ой работы j-ым исполнителем составляет $c_{ij} \geqslant 0$ единиц. Необходимо распределить все работы между исполнителями таким образом, чтобы:

1. Каждый исполнитель исполнял ровно одну работу

2. Суммарная стоимость выполнения всех работ была бы минимальной

Введем управляемые переменные:

$$x_{ij} = \begin{cases} 1, & \text{если } i-\text{ую работу исполняет } j-\text{ый исполнитель} \\ 0, & \text{иначе} \end{cases}$$

Замечание:

Переменную $x_{ij},\ i,j = \overline{1,n}$, удобно записывать в матрицу:

$$x = \begin{bmatrix} x_{11} & \dots & x_{1n} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{nn} \end{bmatrix},$$

которая называется матрицей назначений.

Стоимости c_{ij} записывают в матрицу,

$$C = \begin{bmatrix} C_{11} & \dots & C_{1n} \\ \vdots & & \vdots \\ C_{n1} & \dots & C_{nn} \end{bmatrix},$$

которая называется матрицей стоимостей.

Тогда целевая функция

$$f(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\sum_{i=1}^{n} x_{ij} = 1$$

Условие того, что j-ый исполнитель выполняет ровно одну работу

$$\sum_{j=1}^{n} x_{ij} = 1$$

Условие того, что i-ую работу выполняет ровно один исполнитель

Математическая постановка задачи о назначении:

$$\begin{cases} f(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \to min \\ \sum_{i=1}^{n} x_{ij} = 1, \qquad j=\overline{1,n} \\ \sum_{j=1}^{n} x_{ij} = 1, \qquad i=\overline{1,n} \\ x_{ij} \in \{0,1\}, \qquad i,j=\overline{1,n} \end{cases}$$

Замечание:

Множество допустимых решений задачи о назначении является конечным и состоит из $n!$ элементов. Одним из возможных методов решения этой задачи является прямой перебор допустимых решений однако при больших n он практически не реализуем ввиду большой сложности.

1.2.2 Предварительные соображение о методе решений задачи о назначении:

Соображение 1

Выполним над элементами матрицы стоимостей C следующие преобразования:

1. Из всех элементов $j-$ого столбца вычтем некоторое число α_j, $\qquad j=\overline{1,n}$
2. Из всех элементов $i-$ой строки вычтем некоторое число β_j, $\qquad i=\overline{1,n}$

Обозначимс полученую матрицу $\widetilde{C}$

$$f_{\tilde{C}}(x) = \sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{c}_{ij}x_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n}(c_{ij}-\alpha_j-\beta_i)x_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}x_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_j x_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i x_{ij} = f_c(x) - \gamma$$

$$\gamma = -\sum_{j=1}^{n}\alpha_j - \sum_{i=1}^{n}\beta_i$$

В результате:

$$f_{\tilde{c}}(x) = f_c(x) + const$$

То есть задачи о назначениях с матрицей C и с матрицей $\tilde{C}$ эквивалентно (то есть оптимальные значение функций f_c и $f_{\tilde{c}}$ достигаются при одном и том же x_{ij}.

Соображение 2

Предположим, что в матрице C нашлись n нулей, никакие два их которых не стоят на одной строке и одном столбце.

В этом случае можно сразу записать решение рассматриваемой задачи в матрице x, ставим единицы в тех позициях, в которых в матрице C стоят нули. Остальные элементы матрицы x полагаем равными 0.

$$f(x) = \sum_{i,j} c_{ij}x_{ij} \geqslant 0$$

$$f(x_{opt}) = 0$$

Определение: Набор нулей матрицы С будем называть *системой независимых нулей* (СНН), если никакие 2 нуля этой системы не стоят в одной строке и одном столбце

Замечание: Для решения задачи о назначениях достаточно преобразовать матрицу С к эквивалентному виду, в котором будет содержаться система из n независимых 0.

Замечание: “Слабые места”

1. Возможно после выполнения указанных преобразований мы получим матрицу, из нулей которой в принципе невозможно сформировать систему независимых нулей.

$$\begin{bmatrix} 3 & 1 & 5 & 7 \\ 1 & 1 & 3 & 4 \\ 6 & 1 & 5 & 5 \\ 7 & 1 & 9 & 8 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 0 & 2 & 3 \\ 0 & 0 & 0 & 0 \\ 5 & 0 & 2 & 1 \\ 6 & 0 & 6 & 4 \end{bmatrix}$$

   В этой матрице максимальное число в системе независимых нулей равно 2

2. Возможно в некоторой текущей матрице стоимостей существует набор независимых нулей. Но его построение затруднительно.

В любом из этих случаев построенная система независимых нулей (в которой меньше чем n нулей) может быть улучшена.

Будем считать, что после вычитания наименьших элементов из строк и столбцов матрицы C первоначальная система независимых нулей строится по следующему правилу: просматриваем элементы матрицы C по столбцам в поисках нулей. Если в одном столбце и одной строке с найденным нулем не стоит 0^*, то отмечаем найденые нули звездочкой.

1.2.3 1-ый способ улучшения текущей СНН

Идея: убрать из СНН несколько нулей так, чтобы добавить большее число нулей.

Отметим «+» столбцы, в которых стоят 0^*. Эти столбцы и их элементы будем называть *выделенными.*

Если среди невыделенных элементов есть 0, то можно попытаться улучшить текущую СНН, включив в нее этот 0. Отметим этот 0 штрихом.

В этом случае строка, в которой располагается этот 0, уже не может содержать других элементов СНН. Поэтому если в одной строке с $0'$ есть 0^*, то необходимо снять выделение со столбца, в котором стоит 0^* и выделить строку, в которой стоит $0'$.

Снова среди невыделенных элементов ищем 0 и отмечаем его штрихом. В одной строке с этим $0'$ нет 0^*. Это значит, что можно построить L-цепочку:

от текущего $0' \rightarrow_{\text{по стобцу}} 0^* \rightarrow_{\text{по строке}} 0' \rightarrow 0^* \rightarrow \ldots \rightarrow 0'$

L-цепочка должна быть непродолжаемой и начинаться и заканчиваться $0'$.

В пределах этой L-цепочки 0^* заменяем на просто 0, а $0'$ на 0^*.