

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ “ Информатика и вычислительная техника”
Кафедра “Программное обеспечение вычислительной техники
и автоматизированных систем”

Коледов Л.В.

МЕТОДИЧЕСКОЕ РУКОВОДСТВО для выполнения лабораторных работ по курсу
"Исследование операций"
часть 7.
«Задача о назначениях»

Пособие предназначено для студентов специальностей 010503, 230105

РОСТОВ - НА - ДОНУ

2014

Задание ЛР7. В задании к ЛР6 приведены постановки «ВАШИХ» задач, где надлежит взять матрицы затрат. Выпишите спецификации, необходимые для решения средствами LiPS и MATLAB с применением linprog и bintprog задачу о назначениях при тех же затратах (прибылях) от назначения кандидата A_i на должность B_j . Требуется два варианта расчета – по затратам и прибылям и с применением обоих вариантов linprog и bintprog. Сохраняя протоколы исполнения, сравните оба полученных результата. В тех случаях, когда ВАША матрица не является квадратной, следует дополнить ее до квадратной, добавив колонки или строки произвольным образом.

Пример выполнения. Предложим только варианты решения в MATLAB'е с применением linprog и bintprog. Решение задачи в среде LiPS не содержит принципиальных отличий от версии OpRes06, кроме, акцентуации на том, переменные суть целочисленные.

Вот как выглядит запись скрипта, призванного сгенерировать матрицу выигрышей произвольного размера $m \times n$, построить матриц ограничений и решить задачу о назначениях в среде MATLAB'а обоими методами. Дополнительные операторы позволяют сравнить результаты расчетов по двум вариантам.

```
% Соорудим матрицу ограничений для задачи назначений
% для mm кандидатов на nn должностей
clc
clear all
echo off
diary sign05.txt
lng=5;
S = ones(lng,1);
mm = length(S); % Число кандидатов-строк
T = ones(lng,1);
nn = length(T); % Число должностей-колонок
r1 = [ones(1,mm), zeros(1,mm*(nn-1))]; % mm*nn элементов
% Первая строка матрицы L размера (mm+nn)*[nn*mm]
L2 = [];
L2 = [L2 ; r1]; % Первую строку ставим на место
rr1 = r1;
for rr=2:nn, % Повтор (nn-1) раз
    rr1 = ( mov1(rr1,mm*(nn-1)))';
% Создаем из r1 следующую строку циклическим сдвигом,
% помещающим в начало элемент номер mm*(nn-1)
L2 = [L2; rr1];
end
L1 = []; % создаем L1 - "подвал" матрицы L из nn экземпляров
% единичной матрицы порядка mm
```

```

for ss = 1:nn,
    L1=[L1,eye(mm)];
end
L = [L2;L1];
fm = randsrc(mm,nn,2:11); % Генерируем матрицу прибылей
fm1 = fm'; f = fm1(:); % от назначения
                        % по строкам-поставщикам
lb = zeros(mm*nn,1); % Вектор нижних границ
beq = [S; T]; % Вектор правых частей ограничений
[x, fvalx,exi,out] = bintprog(-f,[],[],L, beq, lb, []);
[y, fvaly,exi,out] = linprog(-f,[],[],L, beq, lb, []);
testx = L*x - beq; % Расхождение с правыми частями x для bintprog
testy = L*y - beq; % Расхождение с правыми частями y для linprog
distsolv = norm(testx - testy); % Расхождение между полученными решениями
compxy = [x y] % Колонки полученных решений
orderx = reshape(x',lng,lng) % Строки - приказы о назначении
ordery = reshape(y',lng,lng) % Строки - приказы о назначении
distf = fvalx - fvaly % Расхождение между
% полученными значениями целевой функции
diary off

```

Вот содержимое файла «sign05.txt».

[?Warning: BINTPROG will be removed in a future release. Use INTLINPROG instead.]?

[?> In bintprog at 93

In tstbigtr04 at 34]?

[?Warning: The given starting point x0 is not binary integer feasible; it will be ignored.]?

[?> In bintprog at 284

In tstbigtr04

at 34]

Optimization terminated.

Optimization terminated.

compxy =

0 0.0000

0 0.0000

0 0.0000

1.0000 1.0000

0 0.0000

0 0.0000

1.0000 1.0000

0 0.0000

0 0.0000

0 0.0000

1.0000 1.0000

0 0.0000

0 0.0000

0 0.0000

0 0.0000

0 0.0000

0 0.0000

1.0000 1.0000

0 0.0000

0 0.0000

0 0.0000

0 0.0000

0 0.0000

0 0.0000

1.0000 1.0000

orderx =

0 0 1 0 0

0 1 0 0 0

0 0 0 1 0

1 0 0 0 0

0 0 0 0 1

ordery =

0.0000 0.0000 1.0000 0.0000 0.0000

0.0000 1.0000 0.0000 0.0000 0.0000

0.0000 0.0000 0.0000 1.0000 0.0000

1.0000 0.0000 0.0000 0.0000 0.0000

0.0000 0.0000 0.0000 0.0000 1.0000

distf =

-3.1910e-08

Приложение

В этом месте приводятся фрагменты из [1, стр. 658-662], необходимые для организации решения задачи ЛП в среде MATLAB и переведенные на русский язык сообщения help-системы MATLAB.

Бинарное линейное программирование

Задача бинарного линейного программирования отличается от общей задачи более жесткими ограничениями на переменные, которые могут принимать только два булевых значения: 0 или 1. Решение задачи в этом случае использует стандартный алгоритм линейного программирования в сочетании с методом ветвей и границ. Чтобы найти оптимальное решение в булевых переменных, строится серия "ослабленных" задач линейного программирования, в которых на переменные (сначала на все) накладываются более слабые ограничения: $0 \leq x(i) \leq 1$.

Алгоритм создает дерево поиска многократным добавлением булевых ограничений. На каждом шаге ветвления алгоритм выбирает некоторую переменную $x(i)$, текущее значение которой не целое, добавляет ограничение $x(i)=0$, чтобы сформировать одну ветку, и ограничение $x(i)=1$, чтобы сформировать альтернативную ветку. Этот процесс можно представить двоичным деревом, в котором каждый узел соответствует введению булева ограничения по одной из переменных.

В каждом узле двоичного дерева алгоритм решает "ослабленную" задачу линейного программирования и выясняет, надо ли строить две ветви от этого узла или переходить к другому узлу. При этом имеется три возможности:

- "ослабленная" задача в текущем узле не имеет решений или значение целевой функции хуже (больше), чем найденное ранее наилучшее булево решение; этот узел удаляется из дерева, и алгоритм переходит к другому узлу в соответствии с некоторой стратегией выбора;
- алгоритм обнаруживает новое допустимое булево решение со значением целевой функции лучшим (меньше), чем найденное ранее наилучшее булево решение; алгоритм корректирует текущее наилучшее решение и переходит на следующий узел;
- "ослабленная" задача в текущем узле дает значение целевой функции лучше (меньше), чем найденное ранее наилучшее булево решение, однако это решение не удовлетворяет булевым ограничениям. В этом случае происходит расширение поиска.

Для решения задачи бинарного линейного программирования в MATLAB 7 используется функция `bintprog`. Формы обращения к этой функции:

```
x = bintprog(f)
x = bintprog(f,A,b)
x = bintprog(f,A,b,Aeq,beq)
x = bintprog(f,A,b,Aeq,beq,x0)
x = bintprog(f,A,b,Aeq,beq,x0,options)
[x,fval] = bintprog(...)
[x,fval,exitflag] = bintprog(...)
[x,fval,exitflag,output] = bintprog(...)
```

Семантика входных и выходных параметров та же, что у функции `linprog`. Через входную переменную `options` с помощью функции `optimset` задаются параметры, управляющие ходом вычислительного процесса и его отображением на мониторе. Отметим некоторые из них, характерные именно для функции `bintprog`:

- ❑ `'MaxIter'` — максимально допустимое число итераций, значение по умолчанию равно `100000*numberOfVariables` (`numberOfVariables` — количество переменных);
- ❑ `'MaxRLPIter'` — максимально допустимое число "ослабленных" задач линейного программирования, значение по умолчанию равно `100*numberOfVariables`;
- ❑ `'MaxNodes'` — максимально допустимое число узлов дерева, значение по умолчанию равно `1000*numberOfVariables`;

Литература.

1. Кетков Ю. Л. И др. MATLAB7: Программирование, численные методы. Гл. 16. СПб.: 2007г.
2. Полисмаков А.И., Математическая экономика. Изд. РГУ. 2006 г.