Решение задач целочисленного, смешанного и булева программирования в среде MATLAB

Г. А. Исаев

ichimaru-gin512@yandex.ru

24 мая 2016 г.

Для решения задач целочисленного, смешанного и булева программирования в среде MATLAB применяется функция `intlinprog`. Работа будет посвящена описанию ее возможностей.

Замечание. Функция `intlinprog` присутствует только в новейших версиях MATLAB (начиная с версии 2014 года).

Функция `intlinprog`

Пусть дана задача смешанного программирования:

$$\begin{gathered} f^T \cdot x \to inf, \\ A \cdot x \leqslant b, \\ Aeq \cdot x = beq, \\ lb \leqslant x \leqslant ub, \end{gathered} \tag{1}$$

где x — вектор, у которого некоторые координаты целочисленны. Для задачи целочисленного программирования все координаты вектора x должны быть целыми, а для задачи булева программирования — должны принимать значения 0 или 1.

Рассмотрим функцию `intlinprog`, решающую эти задачи. Входными данными для этой функции являются:

- вектор коэффициентов целевой функции f;
- матрица ограничений-неравенств A;
- вектор правых частей ограничений-неравенств b;
- множество индексов $intcon$, при которых переменные плана x целочисленны;
- матрица ограничений-равенств Aeq;
- вектор правых частей ограничений-равенств beq;
- вектор lb, ограничивающий план x снизу;
- вектор ub, ограничивающий план x сверху.

На выходе функция `intlinprog` выдает оптимальный план x и минимальное значение целевой функции $fval$.

Пример 1. Решим следующую задачу смешанного программирования в среде MATLAB:

$$\begin{aligned} f(x) = {} & -3x_1 - 2x_2 - x_3 \to inf, \\ & x_1 + \ x_2 + x_3 \leqslant 7, \\ & 4x_1 + 2x_2 + x_3 = 12, \\ & x_3 \in \{0, 1\}, \\ & x_1, x_2 \geqslant 0. \end{aligned}$$

Соответствующая программа (m-файл) будет выглядеть так:

```
clear all
close all
clc % удаляются все текущие переменные из памяти MATLAB, закрываются все графические окна, очищается экран консоли
C = [-3 -2 -1]; % вектор коэффициентов целевой функции
A = [1 1 1]; % матрица ограничений-неравенств, строки матрицы разделяются точкой с запятой (в данном случае состоит из одной строки)
b = [7]; % вектор правых частей ограничений-неравенств
Aeq = [4 2 1]; % матрица ограничений-равенств (в данном случае состоит из одной строки)
beq = [12]; % вектор правых частей ограничений-равенств
lb = zeros(3,1); % нулевой вектор, ограничивающий план x снизу
ub = [Inf Inf 1]; % вектор, ограничивающий план x сверху (число Inf означает бесконечность)
f = C;
intcon = 3; % целочисленный индекс
[x,fval] = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub);
x
fval
```

В результате работы программы получим:

```
Optimal solution found.
x =
     0
     5.5000
     1.0000
fval =
     -12.0000
```

Если в задаче отсутствует какой-то из параметров, то на его место необходимо поставить квадратные скобки `[]`, за исключением случая, когда это последний параметр в списке. Например, если нужно решить задачу без ограничений-равенств и с двусторонними ограничениями на переменные, то оператор вызова функции `intlinprog` будет выглядеть так:

```
[x,fval] = intlinprog(f,intcon,A,b,[],[],lb,ub);
```

Входной параметр *intcon* может быть не только числом, но и массивом или вектором. Чтобы задать, к примеру, индексы 1, 2 и 7, переменные с которыми будут целочисленными, нужно записать в программу строку `intcon = [1 2 7];`. Если нужно задать несколько идущих подряд индексов, например, от 1 до 4, то надо записать строку `intcon = 1:4;`. Для решения задачи целочисленного программирования от n переменных необходимо задать значение *intcon* как массив от 1 до n.

Функция `intlinprog` также способна решать задачи булева программирования. Для этого необходимо объявить все переменные целочисленными и наложить сверху и снизу ограничения, равные 0 и 1 соответственно.

Пример 2. Решим следующую задачу булева программирования в среде MATLAB:

$$\begin{aligned}
f(x) = &-3x_1 - x_2 + 5x_3 - 3x_4 \to inf,\\
&x_1 - 3x_2 - 4x_3 + 5x_4 \leqslant 9,\\
&2x_1 - 4x_2 \leqslant 3,\\
&-9x_3 + x_4 \leqslant 7,\\
&4x_1 + x_3 + 2x_4 = 6,\\
&x_i \in \{0,1\},\ i \in 1:4.
\end{aligned}$$

Соответствующая программа будет выглядеть так:

```
clear all
close all
clc
C = [-3 -1 5 -3];
A = [1 -3 -4 5;2 -4 0 0;0 0 -9 1];
b = [9 3 7];
Aeq = [4 0 1 2];
beq = [6];
intcon = 1:4;
lb = zeros(4,1);
ub = ones(4,1); % задаётся вектор длины четыре, состоящий из единиц
f = C;
[x,fval] = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub);
x
fval
```

Запустив программу, получим сообщение:

```
Optimal solution found.
x =
     1
     1
     0
     1
fval =
     -7
```

MATLAB для решения задачи целочисленного, смешанного и булева программирования использует особую стратегию, которая будет приведена ниже. Если на каком-то шаге стратегии было найдено оптимальное решение, то `intlinprog` завершает свою работу.

Приведём стратегию решения задачи смешанного, целочисленного и булева программирования.

1. Уменьшение размерности задачи, чтобы определить, выполнима ли задача или нет. Также уменьшение размерности упрощает нахождение решения исходной задачи.

2. Решение исходной задачи как задачи линейного программирования[1] (не обращаем внимания на целочисленность).

3. Анализ линейных неравенств $Ax \leqslant b$, чтобы узнать, выполнима ли задача, можно ли убрать лишние ограничения, можно ли усилить ограничения и т.д.

4. Генерация сечений, то есть добавление к задаче дополнительных линейных ограничений, чтобы приблизить решение задачи к целочисленному. Всего есть три набора сечений: базовый, промежуточный и продвинутый. К примеру, метод Гомори принадлежит набору базовых сечений.

5. Использование эвристических методов для нахождения планов задачи. Эти методы базируются на решении задачи линейного программирования, которое было найдено из шага 2. Есть три эвристических метода: округление решения задачи ЛП, нахождение соседних с этим решением планов и совмещение двух вышеперечисленных методов.

6. Использование метода ветвей и границ. Значение целевой функции задачи ЛП является нижней границей для целевой функции исходной задачи. Берется целочисленная переменная x_{i_0}, значение которой в задаче ЛП дробное. Делим исходную задачу на две подзадачи:

   (a) $x_{i_0} \leqslant \lfloor x_{i_0} \rfloor$;
   (b) $x_{i_0} \geqslant \lfloor x_{i_0} \rfloor + 1$.

   Далее берется другая целочисленная переменная x_{i_1}, значение которой в задаче ЛП дробное. Таким же образом делим каждую подзадачу еще на две подзадачи. И так далее.

   Этот метод гарантированно выдаст решение задачи или сообщит, что планов нет.

Функция `intlinprog` позволяет задать настройки с помощью параметра `options`. В частности, можно задать метод решения задачи, максимальное число итераций и т.д. Рассмотрим наиболее существенные опции. Остальные опции можно найти в MATLAB Help. Чтобы открыть MATLAB Help, необходимо запустить сам MATLAB и нажать кнопку «Help», которая находится во вкладке «Home». В открывшемся окне в строке поиска нужно ввести название функции или ключевые слова, которые описывают функцию.

Опция `BranchingRule` задает одно из трех правил выбора вершины в дереве поиска.

- `'maxpscost'` — выбирается вершина, которая максимизирует нижнюю границу целевой функции (используется по умолчанию).

- `'mostfractional'` — выбирается вершина, чья дробная часть близка к 1/2.

[1] Далее в качестве сокращения будем использовать аббревиатуру ЛП.

- `'maxfun'` — выбирается вершина, которая максимизирует абсолютное значение целевой функции.

Чтобы установить правило выбора для максимизации абсолютного значения целевой функции, нужно написать

```
options = optimoptions(@intlinprog,'BranchingRule','maxfun');
[x,fval,exitflag] = intlinprog(f,A,b,Aeq,beq,lb,ub,options);
```

Выходной параметр `exitflag` отвечает за то, как завершилось решение задачи. Приведем таблицу значений этого параметра.

Значение `exitflag`	Интерпретация
2	Работа `intlinprog` была преждевременно остановлена (превышено максимальное время выполнения, максимальное количество проверяемых вершин и т.д.). Был найден целочисленный план.
1	Найдено решение задачи.
0	Работа `intlinprog` была преждевременно остановлена (превышено максимальное время выполнения, максимальное количество проверяемых вершин и т.д.). Целочисленных планов найдено не было.
-2	Множество планов задачи пусто.
-3	Целевая функция не ограничена снизу на множестве планов.

Также можно задать максимальное количество итераций в задаче линейного программирования с помощью опции `LPMaxIter`. По умолчанию оно равно $3 \cdot 10^4$. Чтобы установить допустимое число итераций равным, к примеру, 100, нужно написать:

```
options = optimoptions(@intlinprog,'BranchingRule','maxfun','LPMaxIter',100);
[x,fval,exitflag] = intlinprog(f,A,b,Aeq,beq,lb,ub,options);
```

Если после выполнения сотой итерации решение не будет найдено, то параметр `exitflag` станет равным 0, и на экране появится сообщение:

```
Solver stopped prematurely. No integer feasible point found.
Intlinprog stopped because it exceeded the iteration limit at a node,
options.LPMaxIter = 100 (the selected value).
```

Также можно наложить ограничения на максимальное количество проверяемых вершин (опция `MaxNodes`), на максимальное количество планов (опция `MaxNumFeasPoints`) и на максимальное время выполнения (опция `MaxTime`). По умолчанию стоят следующие установки для этих опций: максимальное количество проверяемых вершин равно 10^7, максимальное время выполнения равно 7200 секундам, а количество планов неограничено. Чтобы задать, например, количество проверяемых вершин равным 200, время выполнения — 300 секундам и максимальное количество планов — 10, нужно написать в программу следующие строки:

```
options = optimoptions(@intlinprog,'MaxNodes',200,'MaxTime',300,
'MaxNumFeasPoints',10);
[x,fval,exitflag] = intlinprog(f,A,b,Aeq,beq,lb,ub,options);
```

Если один из этих параметров будет превышен, то параметр exitflag станет равным 0, и на экране появится одно из сообщений:

```
Intlinprog stopped because it reached the maximum number of nodes,
options.MaxNodes = 200 (the selected value).
```

```
Intlinprog stopped because it exceeded the time limit,
options.MaxTime = 300 (the selected value).
```

```
Intlinprog stopped because it reached the maximum number of feasible points,
options.MaxNumFeasPoints = 10 (the selected value).
```

Кроме того, можно задать набор методов отсечений с помощью опции `CutGeneration`. Каждый метод отсечения добавляет в исходную задачу несколько линейных ограничений. Они нужны для того, чтобы достичь целочисленности в тех целочисленных переменных, которые в решении задачи ЛП имеют дробное значение.

Есть четыре варианта опции `CutGeneration`:

- `'none'` — не задается ни один набор, сечения не будут генерироваться;
- `'basic'` — задается базовый набор сечений (используется по умолчанию);
- `'intermediate'` — задается промежуточный набор сечений;
- `'advanced'` — задается продвинутый набор сечений.

Чтобы установить продвинутый набор сечений, нужно написать

```
options = optimoptions(@intlinprog,'CutGeneration','advanced');
[x,fval,exitflag] = intlinprog(f,A,b,Aeq,beq,lb,ub,options);
```

Выходной параметр `output` содержит информацию о процессе оптимизации, в частности, число проверенных вершин (`numnodes`) и разность между верхней и нижней границей целевой функции (`absolutegap`). Другие поля параметра `output` описаны в MATLAB Help. Запустим с данными из примера 2 следующую программу:

```
options = optimoptions(@intlinprog,'CutGeneration','advanced');
[x,fval,exitflag,output] = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub,options);
exitflag
output.numnodes
output.absolutegap
```

На выходе получим следующее сообщение:

```
exitflag = 1
ans = 0
ans = 0
```

Пример 3. Решим следующую задачу целочисленного программирования в среде MATLAB:

$$\begin{aligned} f(x) = 12x_1 + \ \ 9x_2 + \ \ 2x_3 &\to inf, \\ 10x_1 + 10x_2 + 12x_3 &\leqslant\ 138, \\ 6x_1 + \ \ 8x_2 + 13x_3 &\leqslant\ 67, \\ 7x_1 + \ \ 2x_2 + \ \ 4x_3 &=\ 15, \\ x_1, x_2, x_3 &\text{ — целые числа.} \end{aligned}$$

Соответствующая программа будет выглядеть так:

```
clear all
close all
clc
C = [12 9 2];
A = [10 10 12;6 8 13];
b = [138 67];
Aeq = [7 2 4];
beq = [15];
f = C;
intcon = 1:3;
options = optimoptions(@intlinprog,'BranchingRule','maxfun');
[x,fval,exitflag] = intlinprog(f,intcon,A,b,Aeq,beq,[],[],options);
x
fval
exitflag
```

В результате работы программы получим:

```
Intlinprog stopped because the root LP problem is unbounded.
x =
     []
fval =
     []
exitflag =
     -3
```

Целевая функция не ограничена снизу на множестве планов.

Пример 4. Решим следующую задачу смешанного программирования в среде MATLAB:

$$
\begin{aligned}
f(x) = 8x_1 + \ \ x_2 &\to inf,\\
x_1 + 2x_2 &\geqslant \ -14,\\
-4x_1 - \ \ x_2 &\leqslant \ -33,\\
2x_1 + \ \ x_2 &\leqslant \ 20,\\
x_2 \text{ — целое число.}&
\end{aligned}
$$

Соответствующая программа будет выглядеть так:

```
clear all
close all
clc
C = [8 1];
intcon = 2;
A = [-1 -2;-4 -1;2 1];
b = [14;-33;20];
f = C;
[x,fval,exitflag] = intlinprog(f,intcon,A,b);
x
fval
exitflag
```

В результате работы программы получим:

```
Optimal solution found.
x =
     6.5000
     7.0000
fval =
    59.0000
exitflag =
     1
```

Найдено решение задачи.

Пример 5. Решим следующую задачу смешанного программирования в среде MATLAB:

$$
\begin{aligned}
f(x) = \quad & 12x_1 + 17x_2 + \ \ 9x_3 \to inf, \\
& -2x_1 + 11x_2 + \ \ 5x_3 \leqslant \ -5, \\
& 4x_1 - 10x_2 + 13x_3 \leqslant \ 9, \\
& x_1, x_3 \in \{0, 1\}, \\
& x_2 \geqslant 0.
\end{aligned}
$$

Соответствующая программа будет выглядеть так:

```
clear all
close all
clc
C = [12 17 9];
A = [-2 11 5;4 -10 13];
b = [-5 9];
intcon = [1 3];
lb = zeros(3,1);
ub = [1 Inf 1];
f = C;
[x,fval,exitflag] = intlinprog(f,intcon,A,b,[],[],lb,ub);
x
fval
exitflag
```

В результате работы программы получим:

```
Intlinprog stopped because no point satisfies the constraints.
x =
     []
fval =
     []
exitflag =
     -2
```

Множество планов этой задачи пусто.

Список литературы

[1] MATLAB Help «intlinprog». Информация о функции присутствует в версиях MATLAB с 2014 г.

[2] Кетков Ю. Л., Кетков А. Ю., Шульц М. М. «MATLAB 7: программирование, численные методы». СПб.: БХВ-Петербург, 2005. §16.2, 658–662 сс.