|  |
| --- |
| **Решение задачи линейного программирования** |

http://matlab.exponenta.ru/optimiz/book_4/6/image2118.gif такие, что 

где *f, x, b, beq, lb* и *ub* есть векторы и *A* и *Aeq* есть матрицы.

*Синтаксис:*

x = linprog(f,A,b,Aeq,beq)  
x = linprog(f,A,b,Aeq,beq,lb,ub)  
x = linprog(f,A,b,Aeq,beq,lb,ub,x0)  
x = linprog(f,A,b,Aeq,beq,lb,ub,x0,options)  
[x,fval] = linprog(...)  
[x,fval,exitflag] = linprog(...)  
[x,fval,exitflag,output] = linprog(...)  
[x,fval,exitflag,output,lambda] = linprog(...)

*Описание:*

Linprog решает задачу линейного программирования.

* x = linprog(f,A,b) находит min f'\*x при условии, что A\*x <= b.
* x = linprog(f,A,b,Aeq,beq) решает указанные выше задачу при условии дополнительного выполнения ограничений в виде равенств Aeq\*x = beq. Если нет неравенств, то устанавливается A=[] и b=[].
* x = linprog(f,A,b,Aeq,beq,lb,ub) определяет набор нижних и верхних границ для проектируемых переменных х, так что решение всегда находится в диапазоне lb <= x <= ub. Если нет неравенств, то устанавливается Aeq=[] и beq=[].
* x = linprog(f,A,b,Aeq,beq,lb,ub,x0) устанавливает начальную точку как х0. Эта опция имеет место только для средне-масштабного алгоритма (options.LargeScale равна 'off'). Принимаемый по умолчанию крупно-масштабный алгоритм игнорирует любую стартовую точку.
* x = linprog(f,A,b,Aeq,beq,lb,ub,x0,options) проводит оптимизацию с определенными в структурной опции параметрами оптимизации.
* [x,fval] = linprog(...) возвращает значение целевой функции fun как решение от х: fval = f'\*x.
* [x,lambda,exitflag] = linprog(...) возвращает значение exitflag, которое содержит описание выходных условий.
* [x,lambda,exitflag,output] = linprog(...) возвращает структурный выход с информацией об оптимизации
* [x,fval,exitflag,output,lambda] = linprog(...) Возвращает структурную lambda, чьи поля включают в себя множители Лагранжа как решение от х.

*Аргументы:*

*Входные аргументы.*

Таблица 4-1, Входные аргументы, содержит общее описание аргументов, передаваемых в linprog. Данный подраздел приводит функционально-специфические детали для параметров options

*Выходные аргументы.*

Таблица 4-2. “Выходные аргументы” содержат общее описание возвращаемых linprog аргументов. В этом разделе приводятся общие специфические детали для величин exitflag, lambda и output:

|  |  |
| --- | --- |
| **exitflag** | Описывает выходные условия.   * > 0 Данная функция сходится к решению по х * 0 Максимальное число оценки функции или итерации было превышено * < 0 Функция не сходится к некому решению |
| **lambda** | Структура, которая содержит множители Лагранжа при решении по х (разделенном по типам условий). Поле структуры:   * lower Нижние границы lb * upper Верхние границы ub * ineqlin Линейные неравенства * eqlin Линейные равенства |
| **output** | Структура, которая содержит информацию об оптимизации. Поле структуры:   * iterations Число выполненных итераций * algorithm Используемый алгоритм * cgiterations Число PCG итераций (только для крупно-масштабного алгоритма) |

*Options:*

Параметры оптимизационных опций, используемых в linprog. Часть параметров используется ко всем алгоритмам, некоторые используются только для крупномасштабного алгоритма, а другие применимы только среднемасштабных алгоритмов. Можно использовать optimset для того, чтобы установить или изменить значения данных полей в структуре параметров опций. Для детальной информации смотри Таблицу 4-3, Параметры Опций Оптимизации

LargeScale В случае установки 'on' используется крупно-масштабный, если это возможно, алгоритм. Для использования средне-масштабного алгоритма устанавливается 'off'.

**Medium-Scale и Large-Scale Algorithms.** Эти параметры используются как для средне-масштабного, так и крупно-масштабного алгоритмов.

|  |  |
| --- | --- |
| **Diagnostics** | Проводится печать диагностической информации о минимизируемой функции. |
| **Display** | Уровень отображения. 'off' отображение не производится, 'iter' отображение проводится на каждой итерации, 'final' (принимается по умолчанию) отображение только конечной информации. В настоящее время уровень 'iter' используется только для крупно-масштабного алгоритма. |
| **MaxIter** | Максимальное число допустимых итераций |

**Large-Scale Algorithm Only.** Эти параметры используются только для крупно-масштабного алгоритма:

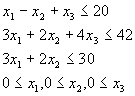
|  |  |
| --- | --- |
| **TolFun** | Конечное допустимое отклонение по значению функции |

*Примеры:*

Найти такое х, что является минимумом от

http://matlab.exponenta.ru/optimiz/book_4/6/image2120.gif

при условии, что



Сперва введем коэффициенты

f = [-5; -4; -6]  
A = [1 -1  1  
    3    2    4  
    3    2    0];  
b = [20; 42; 30];  
lb = zeros(3,1);

Далее обратимся к программе линейного программирования

[x,fval,exitflag,output,lambda] = linprog(f,A,b,[],[],lb);

Вводя x, lambda.ineqlin и lambda.lower получим

x =   
    0.0000  
    15.0000  
    3.0000

lambda.ineqlin =  
    0  
    1.5000  
    0.5000

lambda.lower =  
    1.0000  
    0  
    0

Ненулевые элементы векторов в полях lambda указывают на активные ограничения при решении. В нашем случае, второе и третье ограничения в виде неравенств (в lambda.ineqlin) и первое нижнее граничное ограничение (в lambda.lower) являются активными ограничениями (т.е. решение находится на ограничительных условиях).

*Алгоритм:*

**Крупно-масштабная оптимизация.** Крупно-масштабный метод основан на LIPSOL (Линейное Решение для Внутренней Точки, [3]), которое является вариантом алгоритма предиктор-корректор Меротра ([2]), метод одновременного решения прямой и двойственной задач с внутренней точкой. Перед тем, как алгоритм начнет итерации, выполняется ряд подготовительных шагов. Смотри Крупно-Масштабное Линейное Программирование.

**Средне-масштабная оптимизация.** Linprog использует основанный на алгоритме квадратичного программирования метод проекций. linprog имеет активный набор методов и, таким образом, является вариантом хорошо известного симплексного метода для линейного программирования [1]. Он находит начальное допустимое решение путем первоначального решения задачи линейного программирования.

*Диагностика:*

К**рупно-масштабная оптимизация.**На первой стадии алгоритм может включать некоторую предподготовку ограничений (смотри Крупно-Масштабное Линейное Программирование в разделе “Крупно-Масштабный Алгоритм”). При этом может иметь место несколько ситуаций, когда linprog выдает сообщения о недопустимости. В каждом случае возвращаемый в linprog аргумент exitflag устанавливается в отрицательное значение, что указывает на ошибку.

Если в строке Aeq определены все нули, а соответствующий элемент beq не равен нулю, то выходное сообщение будет

Выход вследствие недопустимости: Полностью нулевая строка для ограничений матрица не имеет нуля в соответствующем компоненте размера с правой стороны

Если для одного из элементов х обнаружено, что он не ограничен снизу, то выходное сообщение будет

Выход вследствие недопустимости: Целевая функция f'\*x не ограничена снизу

Если одна из строк Aeq имеет только один ненулевой элемент, то присваиваемое значение для х вызывается как одноэлементное значение. В этом случае значение данной компоненты х может быть вычислено из Aeq и beq. Если вычисленное значение противоречит другому ограничивающему условию, то выходное сообщение будет

Выход вследствие недопустимости: Одноэлементные переменные в ограничениях типа равенств являются недопустимыми.

Если одноэлементная переменная может быть найдена, но решение противоречит верхней и нижней границам, то выходное сообщение будет

Выход вследствие недопустимости: Одноэлементные переменные в ограничениях типа равенств не находится в пределах границ.

|  |
| --- |
| **Примечание.** Предпроцессорные шаги являются интегральными. Например, хотя если матрица ограничивающих условий не содержит строки полностью из нулей, то прежде всего, другие предпроцессорные шаги могут получить данную ситуацию. |

Сразу после окончания предпроцессорной подготовки начинает работать итеративный раздел алгоритма до тех пор, пока не встретится какой-нибудь критерий останова. (Относительно большей информации о невязках, двойственной проблеме и связанных критериях останова смотри Крупно-Масштабное Линейное Программирование в разделе “Крупно-Масштабный Алгоритм”). Если невязки растут, вместо того, что бы становиться меньше, или невязки ни растут, ни сжимаются, то, соответственно, проявляется одно из следующих завершающих сообщений.

Одна или более невязок, интервал двойственности или общая относительная ошибка выросли в 100000 раз больше, чем было.

или

Одна или более невязок, интервал двойственности или общая относительная ошибка дали команду останов.

После отображения одного из этих посланий, следует одно из шести сообщений, указывающих, что это является результатом недопустимости двойственной, прямой или обеих задач сразу. Это сообщения различается в зависимости от того, как были обнаружены недопустимость или отсутствие границ.

Проявление недопустимости двойственности (а также прямая неограниченность).   
(Прямая невязка < TolFun.)

Проявление недопустимости прямой задачи (а также неограниченность двойственной).   
(Невязка двойственной < TolFun.)

Проявление недопустимости двойственности (а также прямая неограниченность) поскольку невязка двойственной > sqrt(TolFun). (Прямая невязка < 10\*TolFun.)

Проявление недопустимости прямой задачи (а также неограниченность двойственной) поскольку прямая невязка > sqrt(TolFun). (Невязка двойственной < 10\*TolFun.)

Проявление недопустимости двойственности, а также прямая неограниченность поскольку целевая функция прямой < -1e+10 и целевая функция двойственной < 1e+6.

Проявление недопустимости прямой задачи, а также неограниченность двойственной целевая функция двойственной > 1e+10 и целевая функция прямой > -1e+6.

Обе прямая и двойственная задача проявляют недопустимость.

Отметим что, например, прямая (целевая функция) может быть неограниченной, а невязка прямой, что определяется по выполнению ограничений прямой задачи, может быть небольшой.

**Средне-масштабная оптимизация.** Если задача является недопустимой, то linprog дает предупреждение

Предупреждение. Ограничения чрезмерно жесткие. Нет допустимого решения.

В данном случае linprog дает результат, который является минимумом наиболее худшего нарушения ограничений.

Когда ограничения типа равенств являются противоречивыми, linprog дает

Предупреждение. Ограничения типа равенств являются противоречивыми, нет допустимого решения.

Решения без ограничений привели к предупреждению.

Предупреждение. Решение не имеет ограничений и равно бесконечности, ограничительные условия не являются достаточно допустимыми

В этом случае linprog возвращает значение х, которое удовлетворяет условиям ограничения

*Ограничения:*

**Средне-масштабная оптимизация.** К настоящему времени имеются только уровни отображения 'off' и 'final', по применению параметров отображения, итерационный выход по 'iter' не установлен.

*Смотри также:*

[quadprog](http://matlab.exponenta.ru/optimiz/book_4/6/quadprog.php)

*Литература:*

* [1] Dantzig, G.B., A. Orden, and P. Wolfe, "Generalized Simplex Method for Minimizing a Linear from Under Linear Inequality Constraints," Pacific Journal Math. Vol. 5, pp. 183-195.
* [2] Mehrotra, S., "On the Implementation of a Primal-Dual Interior Point Method," SIAM Journal on Optimization, Vol. 2, pp. 575-601, 1992.
* [3] Zhang, Y., "Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Environment,"
* Technical Report TR96-01, Department of Mathematics and Statistics, University of Maryland, Baltimore County, Baltimore, MD, July 1995.