Exam of Optimization - Duration: 2 hours

Documents: lecture slides + handwritten notes (1 extra page)

December 14, 2017

Tear out page 7 and write your name on it.

1 Graphical resolution

The goal is to minimize the objective function $f(\boldsymbol{x})$ (with $\boldsymbol{x} \in \mathbb{R}^2$) whose level sets are shown in Fig. 1, page 7 (larger values of f outside).

1. Unconstrained minimization. Draw on Fig. 1(a) the iterates and the descent directions (with arrows) obtained using the gradient method, with initial solution $\boldsymbol{x}_0$.

2. Constrained minimization. Now, minimization of f is done over the interior of the quadrangle domain of Fig. 1(b). Draw on Fig. 1(b) the iterates and the descent directions obtained using the active-set method, with initial solution $\boldsymbol{x}_1$.

3. Same question using initial solution $\boldsymbol{x}_2$. Please use another color for readability.

2 Quadratic program

Consider the unconstrained quadratic problem:

$$\min_{x_1,x_2} \left\{ f(x_1, x_2) = x_1^2 - x_2 \right\}$$

4. For a given z, the level-set of f is defined as the curve $L_z = \{(x_1, x_2) : \ f(x_1, x_2) = z\}$. Draw on a figure the level set corresponding to $z = 0$. Similarly, draw the level sets L_z for $z = 2, 1, -1$. Please carefully indicate on your figure the value of z corresponding to each curve.

5. What is the solution to the unconstrained quadratic problem?

Consider now the constrained optimization problem:

$$\min_{x_1,x_2} \left\{ f(x_1, x_2) = x_1^2 - x_2 \right\}$$

subject to:

$$x_1 \geq 0 \tag{1}$$
$$x_2 \geq 0 \tag{2}$$
$$x_1 + x_2 = 6 \tag{3}$$

6. On the same figure as above, draw each of the 3 constraints, and illustrate where is the feasible domain.

7. From the graphical drawing with the level sets, what is the solution of the constrained problem?

8. Solve the constrained problem mathematically by writing the KKT conditions and solving the related system.

3 Cutting stock problem

Your company produces large paper rolls of length 10 meters. However, you are allowed to sale paper rolls of smaller lengths to your clients. For instance, your company receives an order for the supply of:

$n_1 = 100$ paper rolls of length $L_1 = 4.5$ meters.

$n_2 = 95$ paper rolls of length $L_2 = 3.0$ meters.

$n_3 = 275$ paper rolls of length $L_3 = 2.0$ meters.

$n_4 = 120$ paper rolls of length $L_4 = 1.5$ meters.

To avoid paper waste, you would like to find the minimum number of 10 meter paper rolls you will cut into small pieces (of lengths L_1, L_2, L_3, L_4).

9. Observe that there are a finite number of ways to cut a paper roll in pieces.

1. The first way is to cut 2 pieces of length L_1, 0 piece of lengths L_2, L_3 and L_4, corresponding to a total length $\mathbf{2} * L_1 + \mathbf{0} * L_2 + \mathbf{0} * L_3 + \mathbf{0} * L_4 = 9 \leq 10$ (the remaining length, 1 meter, is lost).
2. The second way is to cut 1 piece of length L_1, another of length L_2, another of length L_3, and no piece of length L_4, corresponding to a total length $\mathbf{1} * L_1 + \mathbf{1} * L_2 + \mathbf{1} * L_3 + \mathbf{0} * L_4 = 9.5 \leq 10$.
3. *etc.*

The first two ways are represented in the first two rows of matrix $\boldsymbol{N}$:

$$\boldsymbol{N} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Continue this exhaustive list of ways of cutting a 10-meter paper roll in pieces, and fill the remaining entries of matrix $\boldsymbol{N}$.

10. Matrix $\boldsymbol{N}$ is of size $W \times 4$ where W is the number of ways to cut a paper roll. For $i = 1, \ldots, W$, define x_i as the number of 10-meter paper rolls that will be cut according to way i.

Formulate the problem:

find the minimum number of 10 meter paper rolls to be cut
into small pieces to satisfy the request of the client

as an optimization problem. Define the objective function, the variables, and the constraints.

11. What is the structure of the optimization problem? It is not requested to solve the problem here.

4 Matlab implementation

One wishes to minimize

$$f(x,y) \quad = \quad \left[y - \sin(2x) - \frac{x^2}{10}\right]^2 + \exp\left(\frac{x^2+y^2}{100}\right)$$

with respect to (x,y) over the domain:

$$\begin{aligned} 0 \le x \le 10 \\ 0 \le y \le 10 \\ x + y \ge 4 \end{aligned} \tag{4}$$

The Matlab solver `fmincon` will be used, whose simplified documentation is given in appendix.
The following Matlab implementation of function f is proposed:

```
function [F] = fonc_f(x,y);
F = (y-sin(2*x)-x*x/10).^2+exp((x*x+y*y)/100);
```

This function correctly computes $f(x,y)$. For instance, the call `fonc_f(0,0)` returns 1, and `fonc_f(0,1)` returns `2.0101`.

12. However, the writing of `fonc_f` is not consistent with the use `fmincon`. Explain why.

13. Write a modified version of the Matlab function to make function `fonc_f` consistent with the expected inputs of `fmincon`.

14. Among all the ways to call the solver `fmincon` (see Appendix), which one would your recommend? Please indicate the number of input parameters to the function `fmincon` and write the value of each input parameter (`A, b`, *etc.*).

15. `exitflag` is an output parameter of `fmincon`. Explain what is going on when `exitflag` = 0. How can one improve the accuracy of the solution in such case?

16. The output parameter `lambda` identifies with the set of Lagrange multipliers related to the solution. Explain why this information is useful and how to interpret the values of `lambda` (how many Lagrange multipliers? what are the significant values of the Lagrange multipliers?).

17. $f(x,y) = F_1(x,y)^2 + F_2(x,y)^2$ reads as a sum of squares, with

$$F_1(x,y) = y - \sin(2x) - \frac{x^2}{10}$$

$$F_2(x,y) = \exp\left(\frac{x^2+y^2}{200}\right)$$

So the optimization of f could be carried using least-squares solvers, which might be more effective than `fmincon`. Is this a linear or nonlinear least-square problem? Justify your answer.

18. Calculate the Jacobian of

$$F(x,y) = \begin{bmatrix} F_1(x,y) \\ F_2(x,y) \end{bmatrix}$$

19. The output of the optimization algorithm (`fmincon` or the least-squares solver) is sensitive to the initial solution, and different solutions are found depending on the initial solutions, even with fine tuning of option parameters. Propose a strategy to improve the accuracy of the optimization of $f(x,y)$ over the domain (4).

Appendix: help of function `fmincon`

```
    FMINCON attempts to solve problems of the form:

        min f(x)  subject to:  A*x  <= b, Aeq*x  = beq (linear constraints)
         x                     c(x) <= 0, ceq(x) = 0   (nonlinear constraints)
                               lb <= x <= ub           (bounds)

    b and beq are vectors, A and Aeq are matrices, c(x) and ceq(x) are functions
    that return vectors, and f(x) is a function that returns a scalar. f(x), c(x),
    and ceq(x) can be nonlinear functions.

    fmincon implements four different algorithms: interior point, SQP,  active set,
    and trust region reflective. Choose one via the option Algorithm: for instance,
    to choose SQP, set OPTIONS = optimoptions('fmincon','Algorithm','sqp'), and
    then pass OPTIONS to fmincon.

Syntax

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

Description

    fmincon attempts to find a constrained minimum of a scalar function of
    several variables starting at an initial estimate.

    x = fmincon(fun,x0,A,b) starts at x0 and attempts to find a minimum x to
    the function fun subject to the linear inequalities A*x <= b. x0 can be a
    scalar, vector, or matrix.

    x = fmincon(fun,x0,A,b,Aeq,beq) minimizes fun subject to the linear equalities
    Aeq*x = beq as well as A*x <= b. If no inequalities exist, set A = [] and b = [].
```

```
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub) defines a set of lower and upper bounds
on the design variables, x, so that a solution is found in the range
lb <= x <= ub. If no equalities exist, set Aeq = [] and beq = [].
If x(i) is unbounded below, set lb(i) = -Inf, and if x(i) is unbounded above,
set ub(i) = Inf. If no bounds exist, set lb = [] and/or ub = [].

x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon) subjects the minimization to the
nonlinear inequalities c(x) or equalities ceq(x) defined in nonlcon.

x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options) minimizes with the
the default optimization parameters replaced by values in OPTIONS an argument
created with the OPTIMOPTIONS function. For a list of options accepted by
fmincon refer to the documentation of OPTIMOPTIONS.

[x,fval] = fmincon(fun,x0,...) returns the value of the objective function fun
at the solution x.

[x,fval,exitflag] = fmincon(fun,x0,...) returns a value exitflag that describes
the exit condition of fmincon. Possible values of EXITFLAG and the corresponding
exit conditions are listed below.

All algorithms:
  1  First order optimality conditions satisfied.
  0  Too many function evaluations or iterations.
 -1  Stopped by output/plot function.
 -2  No feasible point found.
Trust-region-reflective, interior-point, and sqp:
  2  Change in X too small.
Trust-region-reflective:
  3  Change in objective function too small.
Active-set only:
  4  Computed search direction too small.
  5  Predicted change in objective function too small.
Interior-point and sqp:
 -3  Problem seems unbounded.

[x,fval,exitflag,output] = fmincon(fun,x0,...) returns a structure output with
information such as total number of iterations, and final objective function
value.

[x,fval,exitflag,output,lambda] = fmincon(...) returns the Lagrange multipliers
at the solution x: lambda.lower for lb, lambda.upper for ub, lambda.ineqlin is
for the linear inequalities, lambda.eqlin is for the linear equalities,
lambda.ineqnonlin is for the nonlinear inequalities, and lambda.eqnonlin is
for the nonlinear equalities.
```

FAMILY NAME:

Given Name:

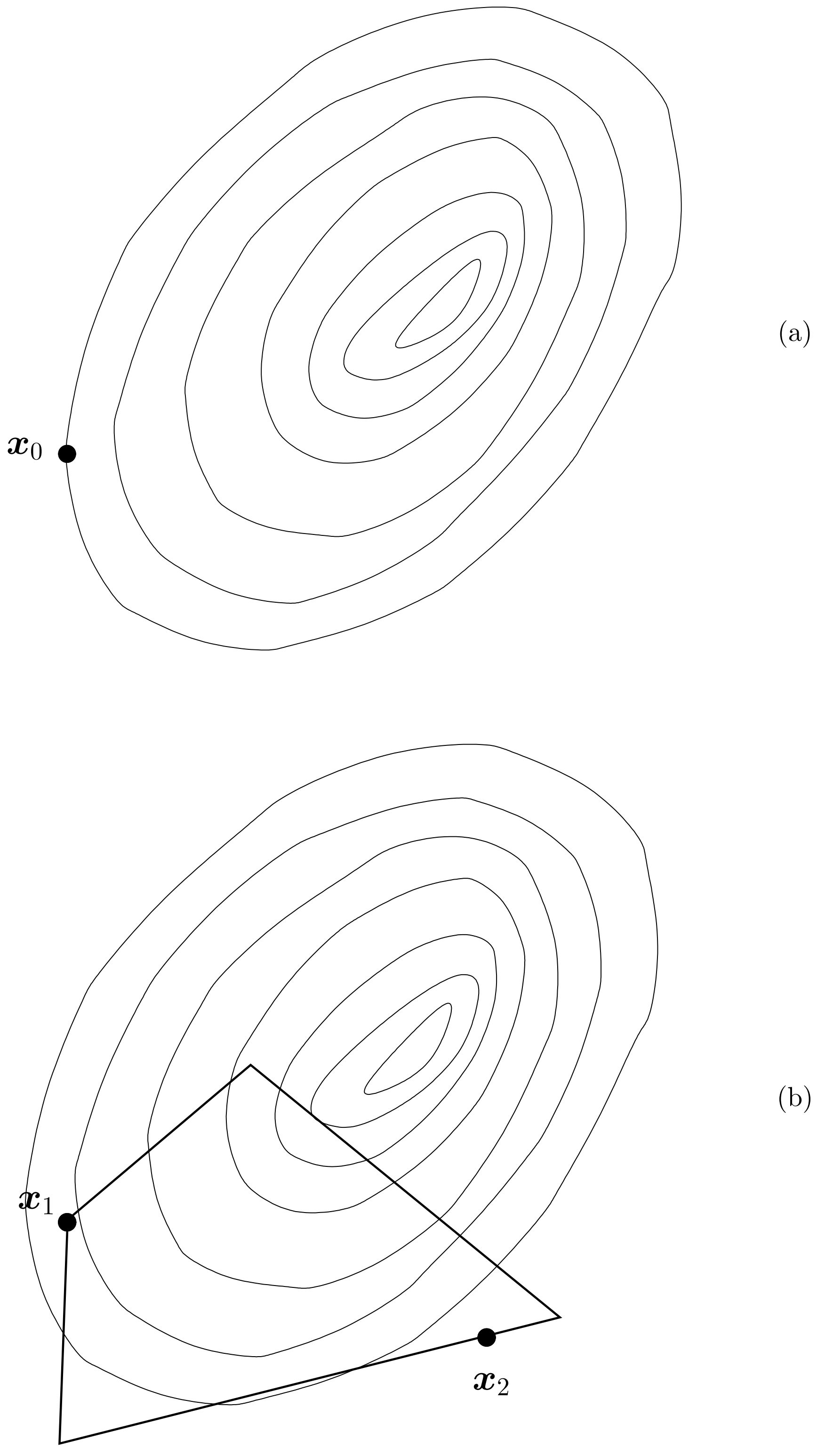

Figure 1: Level sets of the objective function. (a) Unconstrained minimization using the gradient algorithm with initial solution $\boldsymbol{x}_0$. (b) Minimization in the quadrangle domain using the active set algorithm, from the initial solution $\boldsymbol{x}_1$, and from initial solution $\boldsymbol{x}_2$ (use 2 colors).