МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

Нижегородский государственный университет им. Н.И. Лобачевского

Р.С. Бирюков

Методы оптимизации

Учебно-методическое пособие

Рекомендовано методической комиссией факультета ВМК для иностранных студентов, обучающихся в ННГУ по направлению подготовки 010300 «Фундаментальная информатика и информационные технологии» (бакалавриат)

1-е издание

Нижний Новгород

2011

УДК 519.852; 519.853
ББК 22.18
Б-64

Б-64 Бирюков Р.С. МЕТОДЫ ОПТИМИЗАЦИИ: Учебное пособие. – Нижний Новгород: Нижегородский госуниверситет, 2011. – 52 с.

Рецензент: доктор физ.-мат. наук, профессор **Д.В. Баландин**

В настоящем пособии изложены учебно-методические материалы по курсу "Методы оптимизации" для иностранных студентов, обучающихся в ННГУ по направлению подготовки 010300 «Фундаментальная информатика и информационные технологии» (бакалавриат).

Учебно-методическое пособие предназначено для студентов факультета иностранных студентов обучающихся по направлению подготовки 010300 «Фундаментальная информатика и информационные технологии».

УДК 519.852; 519.853
ББК 22.18

Ministry of Education and Science of the Russian Federation

State educational institution of higher education
«Lobachevsky State University of Nizhni Novgorod»

R.S.Biryukov

Method of Optimization

Study book

Recommended by the Methodical Commission of the Faculty of Computer Science
for international students, studying at the B.Sc. programme "010300"

"Fundamental Informatics and Information Technologies"

Nizhny Novgorod

2011

Contents

1. The Linear Programming Problems

Lets start by giving example of linear programming problem.

The Diet Problem
A nutritionist is planning a menu consisting of two main foods A and B. Each ounce of A contains 2 units of fat, 1 unit of carbohydrates, and 4 units of protein. Each ounce of B contains 3 units of fat, 3 units of carbohydrates, and 3 units of protein. The nutritionist wants the meal to provide at least 18 units of fat, at least 12 units of carbohydrates, and at least 24 units of protein. If an ounce of A costs 20 cents and an ounce of B costs 25 cents, how many ounces of each food should be served to minimize the cost of the meal yet satisfy the nutritionist's requirements?

Let x and y denote the number of ounces of foods A and B, respectively, that are served. The number of units of fat contained in the meal is $2x + 3y$, so that x and y have to satisfy the inequality

$$2x + 3y \geq 18.$$

Similarly, to meet the nutritionist's requirements for carbohydrate and protein, we must have x and y satisfy

$$x + 3y \geq 12 \quad \text{and} \quad 4x + 3y > 24.$$

Of course, we also require that $x > 0$ and $y > 0$. The cost of the meal, which is to be minimized, is

$$z = 20x + 25y.$$

Thus, our mathematical model is:

Find values of x and y that will
minimize $z = 20x + 25y$
subject to the restrictions

$$\begin{array}{rcrcr} 2x & + & 3y & \geq & 18 \\ x & + & 3y & \geq & 12 \\ 4x & + & 3y & \geq & 24 \end{array}$$

$$x \geq 0, \quad y \geq 0.$$

Following the form of the previous example, the general linear programming problem can be stated as follows:

Find values of $x_1, x_2, \ldots, x_n$ that will

$$\text{maximize or minimize } z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{1.1}$$

subject to the restrictions

$$\begin{array}{ccccc} a_{11}x_1 & +a_{12}x_2 & +\ldots & +a_{1n}x_n & \leq (\geq)(=)b_1 \\ a_{21}x_1 & +a_{22}x_2 & +\ldots & +a_{2n}x_n & \leq (\geq)(=)b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{m1}x_1 & +a_{m2}x_2 & +\ldots & +a_{mn}x_n & \leq (\geq)(=)b_m \end{array} \tag{1.2}$$

where in each inequality in (1.2) one and only one of the symbols, $<$, $>$, $=$ occurs. The linear function in (1.1) is called the *objective function*. The equalities or inequalities in (2) are called *constraints*. Note that the left-hand sides of all the inequalities or equalities in (2) are linear functions of the variables $x_1, x_2, \ldots, x_n$, just as the objective function is.

We shall say that a linear programming problem is in *standard form* if it is in the following form:

$$\text{maximize} \quad z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{1.3}$$

subject to the restrictions

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & \leq & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & \leq & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & \leq & b_m \end{array} \tag{1.4}$$

$$x_j \geq 0, \quad j = 1, 2, \ldots, n. \tag{1.5}$$

We shall say that a linear programming problem is in *canonical form* if it is in the following form:

$$\text{maximize} \quad z = c_1x_1 + c_2x_2 + \ldots + c_sx_s \tag{1.6}$$

subject to the restrictions

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1s}x_s & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2s}x_s & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{ms}x_s & = & b_m \end{array} \tag{1.7}$$

$$x_j \geq 0, \quad j = 1, 2, \ldots, s. \tag{1.8}$$

We shall now show that every linear programming problem that has unconstrained variables can be solved by solving a corresponding linear programming problem in which all the variables are constrained to be nonnegative. Moreover, we show that every linear programming problem can be formulated as a corresponding standard linear programming problem or as a corresponding canonical linear programming problem. That is, we can show that there is a standard linear programming problem (or canonical linear program problem) whose solution determines a solution to the given arbitrary linear programming problem.

Minimization Problem as a Maximization Problem. Every minimization problem can be viewed as a maximization problem and conversely. This can be seen from the observation that

$$\min \sum_{j=1}^{n} c_j x_j = \max \left(- \sum_{j=1}^{n} c_j x_j \right).$$

Reversing an Inequality. If we multiply the inequality

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \geq b_i$$

by -1, we obtain the inequality

$$-a_{i1}x_1 - a_{i2}x_2 - \ldots - a_{in}x_n \leq -b_i.$$

Changing an Equality to an Inequality. We can write the equation

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n = b_i$$

can be written as the pair of inequalities

$$\begin{aligned} a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n &\leq b_i \\ -a_{i1}x_1 - a_{i2}x_2 - \ldots - a_{in}x_n &\geq -b_i \end{aligned}$$

Unconstrained Variables. Suppose that x_j is not constrained to be nonnegative. We replace x_j with two new variables, x_j^+ and x_j^-, letting

$$x_j = x_j^+ - x_j^-,$$

where $x_j^+ > 0$ and $x_j^- > 0$. That is, any number is the difference of two nonnegative numbers. In this manner we may introduce constraints on unconstrained variables.

We have thus shown that every linear programming problem that is not in standard form can be transformed into an equivalent linear programming problem that is in standard form.

Definition 1.1. A vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ satisfying the constraints of a linear programming problem is called a *feasible solution* to the problem. A feasible solution that minimizes the objective function of a linear programming problem is called an *optimal solution*.

We now describe the method for converting a standard linear programming problem into a problem in canonical form. To do this we must be able to change the inequality constraints into equality constraints. In canonical form the constraints form a system of linear equations, and we can use the methods of linear algebra to solve such systems. In particular, we shall be able to employ the steps used in Gauss-Jordan reduction.

Consider the constraint

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \leq b_i.$$

We may convert it into an equation by introducing a new variable, u_i, and writing

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n + u_i = b_i.$$

The variable u_i is nonnegative and is called a *slack variable*. We now convert the linear programming problem in standard form given by (1.3) and (1.4) to a problem in canonical form by introducing a slack variable in each of the constraints. Note that each constraint will get a different slack variable. In the ith constraint

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \leq b_i,$$

we introduce the slack variable x_{n+i} and write

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n + x_{n+i} = b_i.$$

Because of the direction of the inequality, we know that $x_{n+i} \geq 0$. Therefore, the canonical form of the problem is:

$$\text{maximize} \quad z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{1.9}$$

subject to the restrictions

$$\begin{array}{ccccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & + & x_{n+1} & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & + & x_{n+2} & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & + & x_{n+m} & = & b_m \end{array} \tag{1.10}$$

$$x_j \geq 0, \quad j = 1, 2, \ldots, n+m. \tag{1.11}$$

Moreover, a feasible solution to a standard linear programming problem yields a feasible solution to a canonical linear programming problem by adjoining the values of the slack variables. Conversely, a feasible solution to a canonical linear programming problem yields a feasible solution to the corresponding standard linear programming problem by truncating the slack variables.

Exercises

In Exercises below set up a linear programming model of the situation described. Determine if the model is in standard form. If it is not, state what must be changed to put the model into standard form.

1. A new rose dust is being prepared by using two available products: PEST and BUG. Each kilogram of PEST contains 30 g of carbaryl and 40 g of Malathion, whereas each kilogram of BUG contains 40 g of carbaryl and 20 g of Malathion. The final blend must contain at least 120 g of carbaryl and at most 80 g of Malathion. If each kilogram of PEST costs \$3.00 and each kilogram of BUG costs \$2.50, how many kilograms of each pesticide should be used to minimize the cost?

2. A container manufacturer is considering the purchase of two different types of cardboard-folding machines: model A and model B. Model A can fold 30 boxes per minute and requires 1 attendant, whereas model B can fold 50 boxes per minute and requires 2 attendants. Suppose the manufacturer must fold at least 320 boxes per minute and cannot afford more than 12 employees for the folding operation. If a model A machine costs \$15,000 and a model B machine costs \$20,000, how many machines of each type should be bought to minimize the cost?

3. Dr. R. C. McGonigal treats cases of tactutis with a combination of the brand-name compounds Palium and Timade. The Palium costs \$0.40/pill and the Timade costs \$0.30/pill. Each compound contains SND plus an activator. The typical dosage requires at least 10 mg of SND per day. Palium contains 4 mg of SND and Timade contains 2 mg of SND. In excessive amounts the activators can be harmful. Consequently Dr. McGonigal limits the total amount of activator to no more than 2 mg per day. Palium and Timade each contain 0.5 mg of activator per pill. How many of each pill per day should Dr. McGonigal prescribe to minimize the cost of the medication, provide enough SND, and yet not exceed the maximum permissible limit of activator?

4. A coffee packer blends Brazilian coffee and Colombian coffee to prepare two products: Super and Deluxe brands. Each kilogram of Super coffee contains 0.5 kg of Brazilian coffee and 0.5 kg of Colombian coffee, whereas each kilogram of Deluxe coffee contains 0.25 kg of Brazilian coffee and 0.75 kg of Colombian coffee. The packer has 120 kg of Brazilian coffee and 160 kg of Colombian coffee on hand. If the profit on each kilogram of Super coffee is 20 cents and the profit on each kilogram of Deluxe coffee is 30 cents, how many kilograms of each type of coffee should be blended to maximize profit?

5. Consider an airshed in which there is one major contributor to air pollution — a cement-manufacturing plant whose annual production capacity is 2,500,000 barrels of cement. Figures are not available to determine whether the plant has been operating at capacity. Although the kilns are equipped with mechanical collectors for air pollution control, the plant still emits 2.0 lb of dust per barrel of cement produced. There are two types of electrostatic precipitators that can be installed to control dust emission. The four-field type would reduce emissions by 1.5 lb of dust/barrel and would cost \$0.14/barrel to operate. The five-field type would reduce emissions by 1.8 lb of dust/barrel and would cost \$0.18/barrel to operate. The EPA requires that particulate emissions be reduced by at least 84%. How many barrels of cement should be produced using each new control process to minimize the cost of controls and still meet the EPA requirements?

6. The administrator of a \$200,000 trust fund set up by Mr. Smith's will must adhere to certain guidelines. The total amount of \$200,000 need not be fully invested at any one time. The money may be invested in three different types of securities: a utilities stock paying a 9% dividend, an electronics stock paying a 4% dividend, and a bond paying 5% interest. Suppose that the amount invested in the stocks cannot be more than half the total amount invested; the amount invested in the utilities stock cannot exceed \$40,000; and the amount invested in the bond must be at least \$70,000. What investment policy should be pursued to maximize the return?

7. A book publisher is planning to bind the latest potential bestseller in three different bindings: paperback, book club, and library. Each book goes through a sewing and gluing process. The time required for each process is given in the accompanying table.

	Paperback	Book club	Library
Sewing (min)	2	2	3
Gluing (min)	4	6	10

Suppose the sewing process is available 7 hr per day and the gluing process 10 hr per day. Assume that the profits are \$0.50 on a paperback edition, \$0.80 on a book club edition, and \$1.20 on a library edition. How many books will be manufactured in each binding when the profit is maximized?

8. A local health food store packages three types of snack foods --- Chewy, Crunchy, and Nutty --- by mixing sunflower seeds, raisins, and peanuts. The specifications for each mixture are given in the accompanying table.

Mixture	Sunflower seeds	Raisins	Peanuts	Selling price per kilogram (\$)
Chewy		At least 60%	At most 20%	2.00
Crunchy	At least 60%			1.60
Nutty	At most 20%		At least 60%	1.20

The suppliers of the ingredients can deliver each week at most 100 kg of sunflower seeds at \$1.00/kg, 80 kg of raisins at \$1.50/kg, and 60 kg of peanuts at \$0.80/kg. Determine a mixing scheme that will maximize the store's profit.

9. An agricultural mill manufactures feed for cattle, sheep, and chickens. This is done by mixing the following main ingredients: corn, limestone, soybeans, and fish meal. These ingredients contain the following nutrients: vitamins, protein, calcium, and crude fat. The contents of the nutrients in each kilogram of the ingredients is summarized below.

Ingredient	Nutrient			
	Vitamins	Protein	Calcium	Crude fat
Corn	8	10	6	8
Limestone	6	5	10	6
Soybeans	10	12	6	6
Fish meal	4	8	6	9

The mill contracted to produce 10, 6, and 8 (metric) tons of cattle feed, sheep feed, and chicken feed. Because of shortages, a limited amount of the ingredients is available --- namely, 6 tons of corn, 10 tons of limestone, 4 tons of soybeans, and 5 tons of fish meal. The price per kilogram of these ingredients is respectively \$0.20, \$0.12, \$0.24, and \$0.12. The minimal and

10. A company wishes to plan its production of two items with seasonal demands over a 12-month period. The monthly demand of item 1 is 100,000 units during the months of October, November, and December; 10,000 units during the months of January, February, March, and April; and 30,000 units during the remaining months. The demand of item 2 is 50,000 during the months of October through February and 15,000 during the remaining months. Suppose that the unit product cost of items 1 and 2 is \$5.00 and \$8.00 respectively, provided that these were manufactured prior to June. After June, the unit costs are reduced to \$4.50 and \$7.00 because of the installation of an improved manufacturing system. The total units of items 1 and 2 that can be manufactured during any particular month cannot exceed 120,000. Furthermore, each unit of item 1 occupies 2 cubic feet and each unit of item 2 occupies 4 cubic feet of inventory. Suppose that the maximum inventory space allocated to these items is 150,000 cubic feet and that the holding cost per cubic foot during any month is \$0.10. Formulate the production scheduling problem so that the total production and inventory costs are minimized.

11. Fred has \$2200 to invest over the next five years. At the beginning of each year he can invest money in one- or two-year time deposits. The bank pays 8 percent interest on one-year time deposits and 17 percent (total) on two-year time deposits. In addition, West World Limited will offer three-year certificates at the beginning of the second year. These certificates will return 27 percent (total). If Fred reinvests his money available every year, formulate a linear program to show him how to maximize his total cash on hand at the end of the fifth year.

12. A ten-acre slum in New York City is to be cleared. The officials of the city must decide on the redevelopment plan. Two housing plans are to be considered: low-income housing and middle-income housing. The types of housing can be developed at 20 and 15 units per acre respectively. The unit costs of the low- and middle-income housing are $13,000 and $18,000. The lower and upper limits set by the officials on the number of low-income housing units are 60 and 100. Similarly, the number of middle-income housing units must lie between 30 and 70. The combined maximum housing market potential is estimated to be 150 (which is less than the sum of the individual market limits due to the overlap between the two markets). The total mortgage committed to the renewal plan is not to exceed $2 million. Finally, it was suggested by the architectural adviser that the number of low-income housing units be at least 50 units greater than one-half the number of middle-income housing units. Resolve the problem if the objective is to maximize the number of houses being constructed.

13. A steel manufacturer produces four sizes of I beams: small, medium, large, and extra large. These beams can be produced on any one of three machine types: A, B, and C. The lengths in feet of the I beams that can be produced on the machines per hour are summarized below.

Beam	Machine		
	A	B	C
Small	300	600	800
Medium	250	400	700
Large	200	350	600
Extra large	100	200	300

Assume that each machine can be used up to 50 hours per week and that the hourly operating costs of these machines are respectively $30.00, $50.00, and $80.00. Further suppose that 10,000, 8,000, 6,000, and 6,000 feet of the different-size I beams are required weekly. Formulate the machine scheduling problem as a linear program.

14. A cheese firm produces two types of cheese: Swiss cheese and sharp cheese. The firm has 60 experienced workers and would like to increase its working force to 90 workers during the next eight weeks. Each experienced worker can train 3 new employees in a period of two weeks during which the workers involved virtually produce nothing. It takes one hour to produce 10 pounds of Swiss cheese and one hour to produce 6 pounds of sharp cheese. A work week is 40 hours. The weekly demands (in 1000 pounds) are summarized below.

Cheese type	Week							
	1	2	3	4	5	6	7	8
Swiss cheese	12	12	12	16	16	20	20	20
Sharp cheese	8	8	10	10	12	12	12	12

Suppose that a trainee receives full salary as an experienced worker. Further suppose that overaging destroys the flavor of the cheese so that inventory is limited to one week. How should the company hire and train its new employees so that the labor cost is minimized? Formulate the problem as a linear program.

15. An oil refinery can buy two types of oil: light crude oil and heavy crude oil. The cost per barrel of these types is respectively \$11 and \$9. The following quantities of gasoline, kerosene, and jet fuel are produced per barrel of each type of oil.

	Gasoline	Kerosene	Jet fuel
Light crude oil	0.4	0.2	0.35
Heavy crude oil	0.32	0.4	0.2

Note that 5% and 8% of the crude are lost respectively during the refining process. The refinery has contracted to deliver 1 million barrels of gasoline, 400,000 barrels of kerosene, and 250,000 barrels of jet fuel. Formulate the problem of finding the number of barrels of each crude oil that satisfy the demand and minimize the total cost as a linear program.

16. A company manufactures an assembly consisting of a frame, a shaft, and a ball bearing. The company manufactures the shafts and frames but purchases the ball bearings from a ball bearing manufacturer. Each shaft must be processed on a forging machine, a lathe, and a grinder. These operations require 0.5 hours, 0.2 hours, and 0.3 hours per shaft respectively. Each frame requires 0.8 hours on a forging machine, 0.1 hours on a drilling machine, 0.3 hours on a milling machine, and 0.5 hours on a grinder. The company has 5 lathes, 10 grinders, 20 forging machines, 3 drillers, and 6 millers. Assume that each machine operates a maximum of 2400 hours per year. Formulate the problem of finding the maximum number of assembled components that can be produced as a linear program.

17. A television set manufacturing firm has to decide on the mix of color and black-and-white TV's to be produced. A market research indicates that at most 1000 units and 4000 units of color and black-and-white TV's can be sold per month. The maximum number of man-hours available is 50,000 per month. A color TV requires 20 man-hours and a black-and-white TV requires 15 man-hours. The unit profits of the color and black-and-white TV's are \$60 and \$30 respectively. It is desired to find the number of units of each TV type that the firm must produce in order to maximize its profit. Formulate the problem.

2. The Simplex method

The simplex method is the most widely used method for linear programming and one of the most widely used of all numerical algorithms. It was developed in the 1940's at the same time as linear programming models came to be used for economic and military planning. It had competitors at that time, but these competitors could not match the efficiency of the simplex method and they were soon discarded. Even as problems have become larger and computers more powerful, the simplex method has been able to adapt and remain the method of choice for many people.

In this lecture we describe an elementary version of the simplex method, apply it to a linear program in standard form, show how to find an initial feasible point, and adapt the simplex method to solve degenerate problems.

2.1. The Simplex Method for Problems in Standard Form

Consider a linear programming problem in standard form

$$\text{maximize} \quad z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{2.1}$$

subject to the restrictions

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & \leq & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & \leq & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & \leq & b_m \end{array} \tag{2.2}$$

$$x_j \geq 0, \quad j = 1, 2, \ldots, n. \tag{2.3}$$

We shall now make the additional assumption that $b_i > 0$, $i = 1, \ldots, m$, because when applying the method to these problems it is easy to find a starting point. A procedure for handling problems in which b_i is not nonnegative will be described later.

Transform each of the constraints in (2.2) into an equation by introducing a slack variable we obtain the canonical form of the problem, namely

$$\text{maximize} \quad z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{2.4}$$

subject to the restrictions

$$\begin{array}{ccccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & + & x_{n+1} & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & + & x_{n+2} & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & + & x_{n+m} & = & b_m \end{array} \tag{2.5}$$

$$x_j \geq 0, \quad j = 1, 2, \ldots, n+m. \tag{2.6}$$

Recall that a basic feasible solution to the canonical form of the problem (2.4), (2.5) is an extreme point of the convex set S' of all feasible solutions to the problem.

Definition 2.1. Two distinct extreme points in S' are said to be *adjacent* if as basic feasible solutions they have all but one basic variable in common.

The simplex method is a method that proceeds from a given extreme point (basic feasible solution) to an adjacent extreme point in such a way that the value of the objective function increases or, at worst, remains the same. The method proceeds until we either obtain an optimal solution or find that the given problem has no finite optimal solution. The simplex algorithm consists of two steps: (1) a way of finding out whether a given basic feasible solution is an optimal solution and (2) a way of obtaining an adjacent basic feasible solution with the same or larger value for the objective function. In actual use, the simplex method does not examine every basic feasible solution; it checks only a relatively small number of them. However, examples have been given in which a large number of basic feasible solutions have been examined by the simplex method.

The Initial Basic Feasible Solution. To start the simplex method, we must find a basic feasible solution. The assumption that $b_j \geq 0, j = 1, \dots, m$, allows to take all the nonslack variables as nonbasic variables; that is, we set all the nonslack variables equal to zero. The basic variables are then just the slack variables. We have $x_i = 0$ for $i = 1, \dots, n$ and $x_{n+j} = b_j$ for $j = 1, \dots, m$. This is a feasible solution and it is a basic solution.

We now set up the general problem (2.4), (2.5) and its initial basic feasible solution in tabular form:

	x_1	x_2	$\dots$	x_n	x_{n+1}	x_{n+2}	$\dots$	x_{n+m}	
x_{n+1}	a_{11}	a_{12}	$\dots$	a_{1n}	1	0	$\dots$	0	b_1
x_{n+2}	a_{21}	a_{22}	$\dots$	a_{2n}	0	1	$\dots$	0	b_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
x_{n+m}	a_{m1}	a_{m2}	$\dots$	a_{mn}	0	0	$\dots$	1	b_m
	$-c_1$	$-c_2$	$\dots$	$-c_n$	0	0	$\dots$	0	0

At the top we list the variables $x_1, x_2, \dots, x_n$ as labels on the corresponding columns. The last row, called the *objective row*. Along the left side of each row we indicate which variable is basic in the corresponding equation. Notice that the entry in the last row and rightmost column is the value of the objective function for the initial basic feasible solution.

We are now ready to describe the steps in the simplex method that are used repeatedly to create a sequence of tableaux, terminating in a tableau that yields an optimal solution to the problem.

Checking an Optimality Criterion. We shall now turn to the development of a criterion that will determine whether the basic feasible solution represented by a tableau is optimal. If we write the objective function so that the coefficients of the basic variables are zero, we then have

$$z = \sum_{\text{nonbasic}} d_i x_i + \sum_{\text{basic}} 0 \cdot x_j, \tag{2.7}$$

where the d_j's are the negatives of the entries in the objective row of the tableau. We see that (2.7) has some terms with positive coefficients if and only if the objective row has negative entries under some of the columns labeled by nonbasic variables. Now the value of z can be increased by increasing the value of any nonbasic variable with a negative entry in the objective row from its current value of 0. If this is done, then some basic variable must be set to zero since the number of basic variables is to remain unchanged. Setting this basic variable to zero will not change the value of the objective function since the coefficient of the basic variable was zero. We summarize this discussion by stating the following optimality criterion for testing whether a feasible solution shown in a tableau is an optimal solution.

Optimality Criterion
If the objective row of a tableau has zero entries in the columns labeled by basic variables and no negative entries in the columns labeled by nonbasic variables, then the solution represented by the tableau is optimal.

As soon as the optimality criterion has been met, we can stop our computations, for we have found an optimal solution.

Selecting the Entering Variable. Suppose now that the objective row of a tableau has negative entries in the labeled columns. Then the solution shown in the tableau is not optimal, and some adjustment of the values of the variables must be made.

The simplex method proceeds from a given extreme point (basic feasible solution) to an adjacent extreme point in such a way that the objective function increases in value. From the definition of adjacent extreme point, it is clear that we reach such a point by increasing a single variable from zero to a positive value and decreasing a variable with a positive value to zero. The largest increase in z per unit increase in a variable occurs for the most negative entry in the objective row. If the feasible set is bounded, there is a limit on the amount by which we can increase a variable. Because of this limit, it may turn out that a larger increase in z may be achieved by not increasing the variable with the most negative entry in the objective row. However, this rule is most commonly followed because of its computational simplicity:

$$j^* := \arg\min\{c_j < 0 : j = 1, 2, \ldots, n + m\}$$

The variable x_{j^*} to be increased is called the *entering variable*, since in the next iteration it will become a basic variable; that is, it will enter the set of basic variables. If there are several possible entering variables, choose one.

Choosing the Departing Variable. The variable that goes from basic to nonbasic is called the *departing variable*. The column of the entering variable x_{j^*} is called the *pivotal column*; the row that is labeled with the departing variable is called the *pivotal row*.

We now examine more carefully the selection of the departing variable. Recall that the ratios of the rightmost column entries to the corresponding entries in the pivotal column were determined by how much we could increase the entering variable:

$$\theta_i := b_i / a_{ij^*}, \quad i \in \text{indexes of the basic variables.}$$

These ratios are called *θ-ratios*. The smallest nonnegative θ-ratio is the largest possible value for the entering variable. The basic variable labeling the row where the smallest nonnegative θ-ratio occurs is the departing variable, and the row is the pivotal row.

In the general case, we have assumed that the rightmost column will contain only nonnegative entries. However, the entries in the pivotal column may be positive, negative, or zero. Positive entries lead to nonnegative θ-ratios, which are fine. Negative entries lead to nonpositive θ-ratios. In this case, there is no restriction imposed on how far the entering variable can be increased.

If an entry in the pivotal column is zero, the corresponding θ-ratio is undefined. Hence, in forming the θ-ratios we use only the positive entries in the pivotal column that are above the objective row.

If all the entries in the pivotal column above the objective row are either zero or negative, then the entering variable can be made as large as we wish. Hence, the given problem has no finite optimal solution, and we can stop.

Forming a new tableau. Having determined the entering and departing variables, we must obtain a new tableau showing the new basic variables and the new basic feasible solution.

Step a. Locate and box the entry at the intersection of the pivotal row and pivotal column. This entry is called the *pivot*.

Step b. If the pivot is A, multiply the pivotal row by $1/A$, making the entry in the pivot position in the new tableau equal to 1.

Step c. Add suitable multiples of the new pivotal row to all other rows (including the objective row), so that all other elements in the pivotal column become zero.

Step d. In the new tableau, replace the label on the pivotal row by the entering variable.

These four steps constitute a process called *pivoting*. Steps b and c use elementary row operations and form one iteration of the procedure used to transform a given matrix to reduced row echelon form.

Example 2.1. We solve the following linear programming problem in standard form by using the simplex method:

$$
\begin{array}{l}
\text{maximize} \quad z = 8x_1 + 9x_2 + 5x_3 \\
\text{subject to the restrictions} \\
\begin{array}{rcrcrcr}
x_1 & + & x_2 & + & 2x_3 & \leq & 2 \\
2x_1 & + & 3x_2 & + & 4x_3 & \leq & 3 \\
6x_1 & + & 6x_2 & + & 6x_3 & \leq & 8
\end{array} \\
x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.
\end{array}
$$

To solve this problem by the simplex method, we first convert the problem to canonical form by adding slack variables obtaining

$$
\begin{array}{l}
\text{maximize} \quad z = 8x_1 + 9x_2 + 5x_3 \\
\text{subject to the restrictions} \\
\begin{array}{rcrcrcrcrcrcr}
x_1 & + & x_2 & + & 2x_3 & + & x_4 & & & & & \leq & 2 \\
2x_1 & + & 3x_2 & + & 4x_3 & & & + & x_5 & & & \leq & 3 \\
6x_1 & + & 6x_2 & + & 6x_3 & & & & & + & x_6 & \leq & 8
\end{array} \\
x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.
\end{array}
$$

	x_1	x_2	x_3	x_4	x_5	x_6	
x_4	1	1	2	1	0	0	2
x_5	2	$\boxed{3}$	4	0	1	1	3
x_6	6	6	6	0	0	0	8
	-8	-9	-5	0	0	0	0

	x_1	x_2	x_3	x_4	x_5	x_6	
x_4	$\frac{1}{3}$	0	$\frac{2}{3}$	1	$-\frac{1}{3}$	0	1
x_2	$\frac{2}{3}$	1	$\frac{4}{3}$	0	$\frac{1}{3}$	0	1
x_6	$\boxed{2}$	0	-6	0	-2	1	2
	-2	0	7	0	3	0	9

	x_1	x_2	x_3	x_4	x_5	x_6	
x_4	0	0	$\frac{5}{3}$	1	0	$-\frac{1}{6}$	$\frac{2}{3}$
x_2	0	1	$\frac{10}{3}$	0	1	$-\frac{1}{3}$	$\frac{1}{3}$
x_1	1	0	-3	0	-1	$\frac{1}{2}$	1
	0	0	1	0	1	1	11

Hence, an optimal solution to the standard form of the problem is

$$x_1 = 1, \quad x_2 = \frac{1}{3}, \quad x_3 = 0.$$

The values of the slack variables are

$$x_4 = \frac{2}{3}, \quad x_5 = 0, \quad x_6 = 0.$$

The optimal value of z is 11.

2.2. Degeneracy and Cycling

In choosing the departing variable, we computed the minimum θ-ratio. If the minimum θ-ratio occurs, say, in the rth row of a tableau, we drop the variable that labels that row. Now suppose that there is a tie for minimum θ-ratio, so that several variables are candidates for departing variable. We choose one of the candidates by using an arbitrary rule such as dropping the variable with the smallest subscript. However, there are potential difficulties any time such an arbitrary choice must be made. We now examine these difficulties.

Suppose that the θ-ratios for the rth and sth rows of a tableau are the same and their value is the minimum value of all the θ-ratios. The θ-ratios of these two rows are

$$b_r/a_{rj} = b_s/a_{sj}.$$

Setting the nonbasic variables equal to zero, we find that $x_j = b_r/a_{rj}$ and $x_{i_s} = b_s - a_{sj}b_r/a_{rj} = 0$. Consequently, the tie among the θ-ratios has produced a basic variable whose value is 0.

Definition 2.2. A basic feasible solution in which some basic variables are zero is called *degenerate*.

If no degenerate solution occurs in the course of the simplex method, then the value of z increases as we go from one basic feasible solution to an adjacent basic feasible solution. Since the number of basic feasible solutions is finite, the simplex method eventually stops. However, if we have a degenerate basic feasible solution and if a basic variable whose value is zero departs, then the value of z does not change. In this case the simplex method is said to be cycling and will never terminate by finding an optimal solution or concluding that no bounded optimal solution exists. Cycling can only occur in the presence of degeneracy, but many linear programming problems that are degenerate do not cycle.

Examples of problems that cycle are difficult to construct and rarely occur in practice. However, Kotiah and Steinberg have discovered a linear programming problem arising in the solution of a practical queuing model that does cycle. Also, Beale has constructed the following example of a smaller problem that cycles after a few steps.

Computer programs designed for large linear programming problems provide several options for dealing with degeneracy and cycling. One option is to ignore degeneracy and to assume that cycling will not occur. Another option is to use Bland's Rule for choosing entering and departing variables to avoid cycling:

1. **Selecting the pivotal column.** Choose the column with the smallest subscript from among those columns with negative entries in the objective row instead of choosing the column with the most negative entry in the objective row.
2. **Selecting the pivotal row.** If two or more rows have equal θ-ratios, choose the row labeled by the basic variable with the smallest subscript, instead of making an arbitrary choice.

Bland showed that if these rules are used, then, in the event of degeneracy, cycling will not occur and the simplex method will terminate.

Exercises

1. Use one iteration of the simplex algorithm to obtain the next tableau from the given tableau.

	x_1	x_2	x_3	x_4	
x_4	$\frac{3}{2}$	0	$\frac{5}{3}$	1	6
x_2	$\frac{2}{3}$	1	2	0	8
	-4	0	-2	0	12

2. Use one iteration of the simplex algorithm to obtain the next tableau from the given tableau.

	x_1	x_2	x_3	x_4	
x_1	1	2	0	1	6
x_3	0	$\frac{1}{2}$	1	-1	8
	0	-4	0	-4	$\frac{11}{2}$

3. Solve the linear programming problem using the simplex method.

$$\text{maximize} \quad z = 2x_1 + 3x_2 - 5x_3$$

$$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & \leq & 6 \\ x_1 & - & 3x_2 & - & 3x_3 & \leq & 10 \end{array}$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

4. Solve the linear programming problem using the simplex method.

$$\text{maximize} \quad z = x_1 + 2x_2 + x_3 + x_4$$

$$\begin{array}{rcrcrcrcr} 2x_1 & + & x_2 & + & 3x_3 & + & x_4 & \leq & 8 \\ 2x_1 & + & 3x_2 & & & + & 4x_4 & \leq & 12 \\ 3x_1 & + & x_2 & + & 2x_3 & & & \leq & 18 \end{array}$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0.$$

5. Solve the linear programming problem using the simplex method.

$$\text{maximize} \quad z = 5x_1 + 2x_2 + x_3 + x_4$$

$$\begin{array}{rcrcrcrcr} 2x_1 & + & x_2 & + & x_3 & + & 2x_4 & \leq & 8 \\ 3x_1 & + & & & x_3 & & & \leq & 15 \\ 5x_1 & + & 4x_2 & & & + & x_4 & \leq & 18 \end{array}$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0.$$

6. Solve the linear programming problem using the simplex method.

$$\text{maximize} \quad z = 5x_1 + 2x_2 + x_3 + x_4$$

$$\begin{array}{rcrcrcrcr} 2x_1 & + & x_2 & + & x_3 & + & 2x_4 & \leq & 6 \\ 3x_1 & + & & + & x_3 & & & \leq & 15 \\ 5x_1 & + & 4x_2 & & & + & x_4 & \leq & 24 \end{array}$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0.$$

7. Solve the linear programming problem using the simplex method.

$$\text{maximize} \quad z = -x_1 + 3x_2 + x_3$$

$$\begin{array}{rcrcrcr} -x_1 & - & 2x_2 & - & 7x_3 & \leq & 6 \\ x_1 & + & x_2 & - & 3x_3 & \leq & 15 \end{array}$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

8. Solve the linear programming problem using the simplex method.

$$\text{maximize} \quad z = 3x_1 + 3x_2 - x_3 + x_4$$

$$\begin{array}{rcrcrcrcr} 2x_1 & - & x_2 & - & x_3 & + & x_4 & \leq & 2 \\ x_1 & - & x_2 & + & x_3 & - & x_4 & \leq & 5 \\ 3x_1 & + & x_2 & & & + & 5x_4 & \leq & 12 \end{array}$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0.$$

9. Maximize $z = 6x_1 + 5x_2$ subject to $3x_1 - 2x_2 \leq 0$, $3x_1 + 2x_2 \leq 15$ and $x_1 \geq 0$, $x_2 \geq 0$.

10. Maximize $z = 5x_1 + 4x_2$ subject to $x_1 + 2x_2 \leq 8$, $x_1 - 2x_2 \leq 4$, $3x_1 + 2x_2 \leq 12$, $x_1 \geq 0$, $x_2 \geq 0$.

11. Maximize $z = 3x_1 + 2x_2 + 5x_3$ subject to $2x_1 - x_2 + 4x_3 \leq 12$, $4x_1 + 3x_2 + 6x_3 \leq 18$, $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$.

3. Nonlinear Constrained Optimization

3.1. Conditions for local minimizers

Consider the following optimization problem

$$\min_{\boldsymbol{x}\in D} f(\boldsymbol{x}).$$

The function $f : \mathbb{R}^n \to \mathbb{R}$ that we wish to minimize is a real-valued function, and is called the *objective function*. The vector $\boldsymbol{x} = (x_1, x_2, \dots, x_n)^{\mathrm{T}}$ is an n-vector of independent variables. The set D is a subset of $\mathbb{R}^n$, called the *constraint set* or *feasible set*.

The optimization problem above can be viewed as a decision problem that involves finding the "best" vector $\boldsymbol{x}$ of the independent variables over all possible vectors in D. By the "best" vector we mean the one that results in the smallest value of the objective function. This vector is called the *minimizer* of f over D. It is possible that there may be many minimizers. In this case, finding any of the minimizers will suffice.

There are also optimization problems that require maximization of the objective function. These problems, however, can be represented in the above form because maximizing f is equivalent to minimizing $-f$. Therefore, we can confine our attention to minimization problems without loss of generality.

The above problem is a general form of a constrained optimization problem, because the independent variables are constrained to be in the feasible set D. Often, the feasible set D takes the form $D := \{\boldsymbol{x} : \boldsymbol{h}(\boldsymbol{x}) = 0,\ \boldsymbol{g}(\boldsymbol{x}) \le 0\}$, where $\boldsymbol{h}$ and $\boldsymbol{g}$ are given vector functions. We refer to such constraints as *functional constraints*. If $D = \mathbb{R}^n$, then we refer to the problem as an unconstrained optimization problem. In this lecture, we discuss basic properties of the general optimization problem above, which includes the unconstrained case.

In considering the general optimization problem above, we distinguish between two kinds of minimizers, as specified by the following definitions.

Definition 3.1. Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is a real-valued function defined on some set $D \subset \mathbb{R}^n$. A point $\boldsymbol{x}^* \in D$ is a *local minimizer* of f over D, if there exists $\varepsilon > 0$ such that $f(\boldsymbol{x}) \ge f(\boldsymbol{x}^*)$ for all $\boldsymbol{x} \in D$ and $\|\boldsymbol{x} - \boldsymbol{x}^*\| < \varepsilon$. A point $\boldsymbol{x}^* \in D$ is a *global minimizer* of f over D if $f(\boldsymbol{x}) \ge f(\boldsymbol{x}^*)$ for all $\boldsymbol{x} \in D$.

If, in the above definitions, we replace "$\ge$" with "$>$", then we have a *strict local minimizer* and a *strict global minimizer*, respectively. Notes, that if $\boldsymbol{x}^*$ is a global minimizer than it is also a local minimizer.

Given an optimization problem with constraint set D, a minimizer may lie either in the interior or on the boundary of D. To study the case where it lies on the boundary, we need the notion of feasible directions.

Definition 3.2. Let D be a feasible set, then a vector $\boldsymbol{d} \neq 0$ is a *feasible direction* at point $\boldsymbol{x} \in D$ if there exists $\alpha_0 > 0$ such that $\boldsymbol{x} + \alpha\, \boldsymbol{d} \in D$ for all $\alpha \in [0, \alpha_0]$.

Figure 3.1 illustrates the notion of feasible directions.

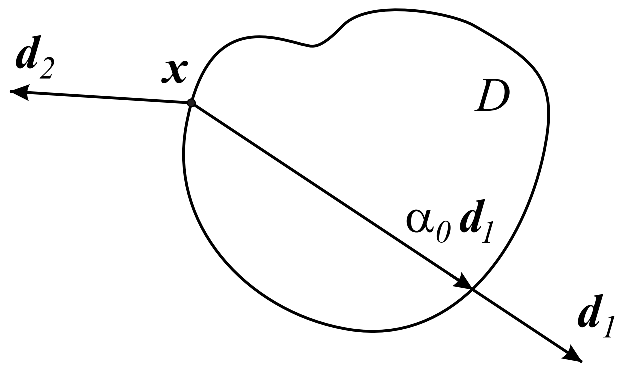

Figure 3.1: Two-dimensional illustration of feasible directions; $\boldsymbol{d}_1$ is a feasible direction, $\boldsymbol{d}_2$ is not a feasible direction

We are now ready to state the following theorem.

Theorem 3.1 (First-Order Necessary Condition). Let D be a subset of $\mathbb{R}^n$ and f a smooth real-valued function on D. If $\boldsymbol{x}^*$ is a local minimizer of f over D, then for any feasible direction $\boldsymbol{d}$ at $\boldsymbol{x}^*$, we have

$$\boldsymbol{d}^{\mathrm{T}} \nabla f(\boldsymbol{x}^*) \geq 0.$$

A special case of interest is when $\boldsymbol{x}^*$ is an interior point of D. In this case, any direction is feasible, and we have the following result.

Corollary 3.1 (Interior case). Let D be a subset of $\mathbb{R}^n$ and f a smooth real-valued function on D. If $\boldsymbol{x}^*$ is a local minimizer of f over D and if $\boldsymbol{x}^*$ is an interior point of D, then

$$\nabla f(\boldsymbol{x}^*) = 0.$$

We now derive a second-order necessary condition that is satisfied by a local minimizer.

Theorem 3.2 (Second-Order Necessary Condition). Let $f \in C^2$ a function on D, $\boldsymbol{x}^*$ a local minimizer of f over D, and $\boldsymbol{d}$ a feasible direction at $\boldsymbol{x}^*$. If $\boldsymbol{d}\mathrm{T}\nabla f(\boldsymbol{x}^*) = 0$, then $\boldsymbol{d}\mathrm{T}H(\boldsymbol{x}^*)\boldsymbol{d} > 0$, where **H** is the Hessian of f.

In the examples below, we show that the necessary conditions are not sufficient.

Example 3.1. Let $f(\boldsymbol{x}) = x_1^2 - x_2^2$. The First-Order Necessary Condition requires that $\nabla f(\boldsymbol{x}) = (2x_1, -2x_2) = \mathbf{0}^{\mathrm{T}}$. Thus, $\boldsymbol{x} = (0,0)^{\mathrm{T}}$ satisfies the First-Order Necessary Condition. The Hessian matrix of f is

$$H(\boldsymbol{x}) = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}.$$

The Hessian matrix is indefinite; that is, for some $\boldsymbol{d}_1 \in \mathbb{R}^2$ we have $\boldsymbol{d}_1^{\mathrm{T}} H(x)\boldsymbol{d}_1 > 0$, e. g., $\boldsymbol{d}_1 = (1,0)^{\mathrm{T}}$; and, for some $\boldsymbol{d}_2$, we have $\boldsymbol{d}_2^{\mathrm{T}} H(x)\boldsymbol{d}_2 < 0$, e. g., $\boldsymbol{d}_2 = (0,1)^{\mathrm{T}}$. Thus, $\boldsymbol{x} = (0,0)^{\mathrm{T}}$ does not satisfy the Second-Order Necessary Condition, and hence it is not a minimizer.

We now derive sufficient conditions that imply that $\boldsymbol{x}^*$ is a local minimizer.

Theorem 3.3 (Second-Order Sufficient Condition, Interior Case). Let $f \in C^2$ be defined on a region in which $\boldsymbol{x}^*$ is an interior point. Suppose that

1. $\nabla f(\boldsymbol{x}^*) = 0$;
2. $H(\boldsymbol{x}^*) > 0$

Then, $\boldsymbol{x}^*$ is a strict local minimizer of f.

Example 3.2. Let $f(\boldsymbol{x}) = x_1^2 + x_2^2$. We have $\nabla f(\boldsymbol{x}) = (2x_1, 2x_2) = \mathbf{0}^{\mathrm{T}}$ if and only if $\boldsymbol{x} = (0,0)^{\mathrm{T}}$. For all $\boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2$, we have

$$H(\boldsymbol{x}) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} > 0.$$

The point $\boldsymbol{x} = (0,0)^{\mathrm{T}}$ satisfies the First-Order Necessary Condition and Second-Order Sufficient Condition. It is a strict local minimizer. Actually $\boldsymbol{x} = (0,0)^{\mathrm{T}}$ is a strict global minimize.

3.2. Karush-Kuhn-Tucker Condition

Now we discuss methods for solving a class of nonlinear constrained optimization problems that can be formulated as:

$$\min\{f(\boldsymbol{x}) : g_i(\boldsymbol{x}) < 0,\ i = 1,\dots,p,\ h_j(\boldsymbol{x}) = 0,\ j = 1,\dots,q\}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g_i : \mathbb{R}^n \to \mathbb{R}$, $h_j : \mathbb{R}^n \to \mathbb{R}$, and $q < n$.

Definition 3.3. Any point satisfying the constraints is called a *feasible point*. The set of all feasible points

$$\{\boldsymbol{x} \in \mathbb{R}^n : g_i(x) < 0,\ i = 1,\dots,p,\ h_i(\boldsymbol{x}) = 0,\ i = 1,\dots,q\}$$

is called the *feasible set*.

We introduce the following definition.

Definition 3.4. An inequality constraint $g_j(\boldsymbol{x}) \le 0$ is said to be *active* at $\boldsymbol{x}^*$ if $g_j(\boldsymbol{x}^*) = 0$. It is *inactive* at $\boldsymbol{x}^*$ if $g_j(\boldsymbol{x}^*) < 0$.

By convention, we consider an equality constraint $h_i(\boldsymbol{x}) = 0$ to be always active.

Definition 3.5. Let $\boldsymbol{x}^*$ satisfy $h(\boldsymbol{x}^*) = 0$, $g(\boldsymbol{x}^*) \le 0$, and let $J(\boldsymbol{x}^*)$ be the index set of active inequality constraints, that is,

$$J(\boldsymbol{x}^*) \triangleq \{j : g_j(\boldsymbol{x}^*) = 0\}.$$

Definition 3.6. A point x^* satisfying the constraints $h_1(\boldsymbol{x}^*) = 0, \ldots, h_q(\boldsymbol{x}^*) = 0$ is said to be a *regular point* of the constraints if the vectors

$$\nabla g_i(\boldsymbol{x}^*),\ \nabla h_j(\boldsymbol{x}^*), 1 < j < q, i \in J(\boldsymbol{x}^*)$$

are linearly independent.

We now state a first-order necessary condition for a point to be a local minimizes. We call this condition the Karush-Kuhn-Tucker condition. In the literature, this condition is sometimes also called the Kuhn-Tucker condition.

Theorem 3.4 (Karush-Kuhn-Tucker Theorem). Let $f, g_i, h_j \in C^2$. Let $\boldsymbol{x}^*$ be a regular point and a local minimizer for the problem of minimizing f subject to $g_i(x) < 0$, $i = 1, \ldots, p$, $h_i(\boldsymbol{x}) = 0$, $i = 1, \ldots, q\}$. Then, there exist $\boldsymbol{\lambda}^* = (\lambda_1, \ldots, \lambda_p)$ and $\boldsymbol{\mu}^* = (\mu_1, \ldots, \mu_q)$ such that

1. $\boldsymbol{\lambda}^* \geq 0$,;
2. $\nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{p} \lambda_i^* \nabla g_i(\boldsymbol{x}^*) + \sum_{j=1}^{q} \mu_j^* \nabla h_j(\boldsymbol{x}^*) = \mathbf{0}^{\mathrm{T}}$;
3. $(\boldsymbol{\lambda}^*)^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x}^*) = 0$.

In the above theorem, we refer to $\boldsymbol{\mu}^* = (\mu_1^*, \ldots, \mu_q^*)$ as the Lagrange multiplier vector, and $\boldsymbol{\lambda}^* = (\lambda_1^*, \ldots, \lambda_p^*)$ as the Karush-Kuhn-Tucker multiplier vector. We refer to their components as Lagrange multipliers and Karush-Kuhn-Tucker multipliers, respectively.

Let us discuss this theorem. Observe that $\lambda_i^* \geq 0$ (by condition 1) and $g_i(\boldsymbol{x}^*) \leq 0$. Therefore, the condition

$$(\boldsymbol{\lambda}^*)^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x}^*) = \lambda_1^* g_1(\boldsymbol{x}^*) + \ldots + \lambda_p^* g_p(\boldsymbol{x}^*) = 0$$

implies that if $g_i(\boldsymbol{x}^*) < 0$, then $\lambda_i^* = 0$, that is, for all $i \notin J(\boldsymbol{x}^*)$, we have $\lambda_i^* = 0$. In other words, the Karush-Kuhn-Tucker multipliers λ_i^* corresponding to inactive constraints are zero. The other Karush-Kuhn-Tucker multipliers, $\lambda_i^* p$, $i \in J(\boldsymbol{x}^*)$, are nonnegative; they may or may not be equal to zero.

We apply the Karush-Kuhn-Tucker condition in the same way we apply any necessary condition. Specifically, we search for points satisfying the KKT condition and treat these points as candidate minimizers. To summarize, the Karush-Kuhn-Tucker condition consists of five parts (three equations and two inequalities):

1. $\lambda_i^* \geq 0, i = 1, 2, \ldots, p$;
2. $\nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{p} \lambda_i^* \nabla g_i(\boldsymbol{x}^*) + \sum_{j=1}^{q} \mu_j^* \nabla h_j(\boldsymbol{x}^*) = \mathbf{0}^{\mathrm{T}}$;
3. $\lambda_i^* g_i(\boldsymbol{x}^*) = 0, i = 1, 2, \ldots, p$;
4. $g_i(\boldsymbol{x}^*) \leq 0, i = 1, 2, \ldots, p$;
5. $h_j(\boldsymbol{x}^*) = 0, j = 1, 2, \ldots, q$.

Exercises

1. Examine the following functions for maxima, minima, and points of inflection, without using the second derivative:

(a) $f(x) = 3x^2 + 12x + 10$;

(b) $f(x) = 3x^2 - 12x + 10$;

(c) $f(x) = -3x^2 + 12x + 10$.

2. Find maxima and/or minima for each of the following functions, using both first and second derivatives. Indicate in each case whether the points are relative or absolute maxima or minima.

(a) $f(x) = 2x^3 + 3x^2 - 12x - 15$;

(b) $f(x) = 3x^4 - x^3 + 2$;

(c) $f(x) = x^3$;

(d) $f(x) = 3x^2 + 12x + 10$;

(e) $f(x) = 8x^3 - 9x^2 + 1$;

(f) $f(x) = x^4 - 18x^2 + 15$.

3. Explore the following functions for maxima, minima, and inflection points:

(a) $f(x) = 0.1(x+1)^3(x-2)^3 + 1$;

(b) $f(x) = (x+3)^3(x-2)^2 + 1$.

4. Find maxima and/or minima for each of the following functions:

(a) $f(x, y) = x^2 + x\,y + y^2 - y$;

(b) $f(x, y) = x^2 + y^2 + (3x + 4y - 26)^2$;

(c) $f(x, y) = 9x^2 - 18x - 16y^2 - 64y - 55$;

(d) $f(x, y) = 16 - 2(x-3)^2 - (y-7)^2$;

(e) $f(x, y) = -2x^3 + 6x\,y - y^2 - 4y + 100$.

5. For each value of the scalar A, find the set of all stationary points of the following function of the two variables x and y

$$f(x, y) = x^2 + y^2 + A\,x\,y + x + 2y.$$

Which of these stationary points are global minima?

6. Show that the function $f(x, y) = (x^2 — 4)^2 + y^2$ has two global minima and one stationary point, which is neither a local maximum nor a local minimum.

7. Show that the function $f(x, y) = (y — x^2)^2 — x^2$ has only one stationary point, which is neither a local maximum nor a local minimum.

8. Find all local minima of the function $f(x, y) = \frac{1}{2}x^2 + x\,\cos y$.

4. One-dimensional search methods

The search methods we discuss in this lecture allow us to determine the minimizer of a function $f : \mathbb{R} \to \mathbb{R}$ over a closed interval, say $[a, b]$. The only property that we assume of the objective function $f(x)$ is that it is unimodal which means that $f(x)$ has only one local minimizer. More precisely,

Definition 4.1. Let $f(x)$ be a function defined on interval $[a, b]$. If there is $x^* \in [a, b]$ such that $f(x)$ is strictly decreasing on $[a, x^*]$ and strictly increasing on $[x^*, b]$, then $f(x)$ is called a *unimodal function* on $[a, b]$. Such an interval $[a, b]$ is called a *unimodal interval* related to $f(x)$.

Note that, first, the unimodal function does not require the continuity and differentiability of the function; second, using the property of the unimodal function, we can exclude portions of the interval of uncertainty that do not contain the minimum, such that the interval of uncertainty is reduced. The following theorem shows that if the function f is unimodal on $[a, b]$, then the interval of uncertainy could be reduced by comparing the function values of f at two points within the interval.

Theorem 4.1. Let $f(x)$ be unimodal on $[a, b]$. Let x_1 and x_2 are two arbitrary points such that $a \le x_1 < x_2 \le b$. Then

1. if $f(x_1) \le f(x_2)$, then $[a, x_2]$ is a unimodal interval related to f;
2. if $f(x_1) \ge f(x_2)$, then $[x_1, b]$ is a unimodal interval related to f.

Now we present several algorithms for minimizing a unimodal function over a closed bounded interval by iteratively reducing the interval of uncertainty.

4.1. Dichotomous Search

Consider a unimodal function $f(x)$ which is known to have a minimizer x^* in the interval $[a_0, b_0]$. Obviously, the smallest number of functional evaluations that is needed to reduce the interval of uncertainty is two. If we consider two points x_1 and x_2, then, by Theorem 4.1, the new interval of uncertainty depends on the values of function f at points x_1 and x_2. Note, however, that we do not know, a priori, whether $f(x_1) < f(x_2)$ or $f(x_1) > f(x_2)$. Thus, the optimal strategy is to place x_1 and x_2 in such a way as to guard against the worst possible outcome, that is, to minimize the maximum of $x_2 - a_0$ and $b_0 - x_1$. This can be accomplished by placing x_1 and x_2 at the midpoint of the interval $[a_0, b_0]$. If we do this, however, we would have only one trial point and would not be able to reduce the interval of uncertainty. Therefore x_1 and x_2 are placed symmetrically, each at a distance $\varepsilon > 0$ from the midpoint. Here, $\varepsilon > 0$ is a scalar that is sufficiently small so that the new length of uncertainty, $(b_0 - a_0)/2 + \varepsilon$, is close enough to the theoretical optimal value of $(b_0 - a_0)/2$ and, in the meantime, would make the ftinctional evaluations $f(x_1)$ and $f(x_2)$ distinguishable.

In dichotomous search, we place each of the first two observations, x_1 and x_2, symmetrically at a distance ε from the midpoint $(a_0+b_0)/2$. Depending on the values of f at x_1 and x_2, a new interval of uncertainty is obtained. The process is then repeated by placing two new observations.

---

begin
  Initialize initial interval of uncertainty $[a_0, b_0]$
  Initialize the distinguishability constant $\varepsilon > 0$
  Initialize tolerance $\delta > 0$
  $k := 0$
  while $b_k - a_k > \delta$
    $x_1 := (a_k + b_k)/2 - \varepsilon$
    $x_2 := (a_k + b_k)/2 + \varepsilon$
    if $f(x_1) < f(x_2)$ **then**
      $a_{k+1} := a_k$
      $b_{k+1} := x_2$
    else
      $a_{k+1} := x_1$
      $b_{k+1} := b_k$
    end
    $k := k + 1$
  end
  $x^* := (a_k + b_k)/2$
end

Figure 4.1: Dichotomous Search Algorithm

Note that the length of uncertainty at the beginning of iteration $k + 1$ is given by

$$b_{k+1} - a_{k+1} = \frac{b_0 - a_0}{2^k} + 2\varepsilon\left(1 - \frac{1}{2^k}\right).$$

This formula can be used to determine the number of iterations needed to achieve the desired accuracy. Since each iteration requires two observations, the formula can also be used to determine the number of observations.

Example 4.1. Use the Dichotomous search method to find the value of x that minimizes

$$f(x) = x^4 - 14x^3 + 60x^2 - 70x$$

in the range $[0, 2]$. Locate this value of x to within a range of 0.3.

Let the distinguishability constant is 0.1, then after N stages the range $[0, 2]$ is reduced to

$$\frac{2}{2^N} + 0.2\left(1 - \frac{1}{2^N}\right).$$

So, we choose N so that $2^{1-N} + 2\varepsilon \leq 0.3$. Five stages of reduction will do; that is, $N = 5$. The first two observations are located at $x_1 = 0.9$, $x_2 = 1.1$. Note that $f(x_1) < f(x_2)$. Hence, the new interval of uncertainty is $[0.0, 1.1]$. The process is repeated, and the computations are summarized in Table 4.1. After five iterations involving ten observations, the interval of uncertainty is $[0.675, 0.931]$, so that the minimum can be estimated to be the midpoint 0.803. Note that the true minimum is in fact 0.781.

Iteration k	a_k	b_k	x_1	x_2	$f(x_1)$	$f(x_2)$
1	0.000	2.000	0.900	1.100	−23.950	−21.570
2	0.000	1.100	0.450	0.650	−20.585	−23.816
3	0.450	1.100	0.675	0.875	−24.011	−24.105
4	0.675	1.100	0.788	0.988	−24.368	−23.146
5	0.675	0.988	0.731	0.931	−24.292	−23.708

Table 4.1: Summary of computations for the Dichotomous search method

4.2. Fibonacci Search

Consider a unimodal function f of one variable and the interval $[a_0, b_0]$. We choose the intermediate points x_1 and x_2 in such a way that, first, an approximation to the minimizer of f may be achieved in as few evaluations as possible and, second, the reduction in the range is symmetric, in the sense that

$$x_1 - a_k = b_k - x_2 = \lambda_k(b_k - a_k).$$

Our goal is to select successive values of λ_k, such that only one new function evaluation is required at each stage. To derive the strategy for selecting evaluation points, consider Figure 4.2. From Figure 4.2, we see that it is sufficient to choose the λ_k such that

$$\lambda_{k+1}(1 - \lambda_k) = 1 - 2\lambda_k.$$

After some manipulations, we obtain

$$\lambda_{k+1} = 1 - \frac{\lambda_k}{1 - \lambda_k} \tag{4.1}$$

where $0 < \lambda_k < 1/2$.

Suppose that we are given a sequence $\lambda_1, \lambda_2, \ldots$ satisfying the above law of formation, and we use this sequence in our search algorithm. Then, after N iterations of the algorithm, the uncertainty range is reduced by a factor of

$$(1 - \lambda_1)(1 - \lambda_2) \ldots (1 - \lambda_N).$$

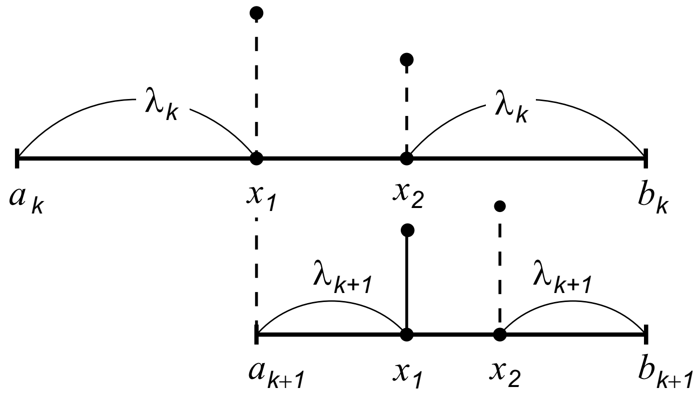

Figure 4.2: Selecting evaluation points

Depending on the sequence $\lambda_1, \lambda_2, \ldots$, we get a different reduction factor. The natural question is as follows: What sequence $\lambda_1, \lambda_2, \ldots$ minimizes the above reduction factor?

Before we give the answer on this question, we first need to introduce the *Fibonacci sequence*, $F_1, F_2, F_3, \ldots$. This sequence is defined as follows. First, let $F_0 = 0$ and $F_1 = 1$ by convention. Then, for $k \geq 0$,

$$F_{k+2} = F_{k+1} + F_k.$$

Some values of elements in the Fibonacci sequence are as follows:

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
1	1	2	3	5	8	13	21	34	55	89

It turns out that the solution to the above optimization problem is:

$$\lambda_k = 1 - F_{N-k+1}/F_{N-k+2},$$

where the F_k are the elements of the Fibonacci sequence. The resulting algorithm is called the *Fibonacci search method*. We skip over a proof for the optimality of the Fibonacci search method.

We point out that there is an anomaly in the final iteration of the Fibonacci search method, because

$$\lambda_N = 1 - \frac{F_1}{F_2} = \frac{1}{2}.$$

Recall that we need two intermediate points at each stage, one that comes from a previous iteration and another that is a new evaluation point. However, with $\lambda_N = 1/2$, the two intermediate points coincide in the middle of the uncertainty interval, and therefore we cannot further reduce the uncertainty range. To get around this problem, we perform the new evaluation for the last iteration using $\lambda_N = 1/2 - \varepsilon$, where ε is a small number. In other words, the new evaluation point is just to the left or right of the midpoint of the uncertainty interval.

begin
Initialize initial interval $[a_0, b_0]$ *and tolerance* $\delta > 0$
Initialize the distinguishability constant $\varepsilon > 0$
Choose N *such that* $(1 + 2\varepsilon)/F_{N+1} \leq \delta$
for $k := 1$ **to** N **do**
$\lambda_k := 1 - F_{N-k+1}/F_{N-k+2}$
if $k = N$ **then**
$\lambda_k := 1/2 - \varepsilon$
end
$x_1 := a_k + \lambda_k(b_k - a_k)$
$x_2 := b_k - \lambda_k(b_k - a_k)$
if $f(x_1) < f(x_2)$ **then**
$a_{k+1} := a_k$
$b_{k+1} := x_2$
else
$a_{k+1} := x_1$
$b_{k+1} := b_k$
end
end
$x^* := (a_k + b_k)/2$
end

Figure 4.3: Fibonacci Search Algorithm

As a result of the above modification, the reduction in the uncertainty range at the last iteration is

$$1 - (\lambda_N - \varepsilon) = \frac{1}{2} + \varepsilon = \frac{1 + 2\varepsilon}{2}.$$

Therefore the reduction factor in the uncertainty range for the Fibonacci method is

$$(1 - \lambda_1)(1 - \lambda_2) \dots (1 - (\lambda_N - \varepsilon)) = \frac{1 + 2\varepsilon}{F_{N+1}}$$

Example 4.2. Use the Fibonacci search method to find the value of x that minimizes

$$f(x) = x^4 - 14x^3 + 60x^2 - 70x$$

in the range $[0, 2]$. Locate this value of x to within a range of 0.3.

After N steps the range is reduced by $(1 + 2\varepsilon)/F_{N+1}$ in the worst case. We need to choose N such that

$$\frac{1 + 2\varepsilon}{F_{N+1}} \leq \frac{0.3}{2} = 0.15$$

If we choose $\varepsilon = 0.1$, then $N = 5$ will do. The first two observations are located at $x_1 = 0.9$, $x_2 = 1.1$. Note that $f(x_1) < f(x_2)$. Hence, the new interval of uncertainty is $[0.00, 1.25]$. The process is repeated, and the computations are summarized in Table 4.2. After five iterations involving ten observations, the interval of uncertainty is $[0.75, 0.90]$, so that the minimum can be estimated to be the midpoint 0.825. Note that the true minimum is in fact 0.781.

Iteration k	a_k	b_k	x_1	x_2	$f(x_1)$	$f(x_2)$
1	0.000	2.000	0.750	1.250	-24.340	-18.652
2	0.000	1.250	0.500	0.750	-21.688	-24.340
3	0.500	1.250	0.750	1.000	-24.340	-23.000
4	0.500	1.000	0.750	0.750	-24.340	-24.340
5	0.750	1.000	0.850	0.900	-24.226	-23.950

Table 4.2: Summary of computations for the Fibonacci Search method

4.3. Golden Section Search

The main disadvantage of the Fibonacci search is that the number of iterations must be supplied as input. A search method in which iterations can be performed until the desired accuracy in either the minimizer or the minimum value of the objective function is achieved is the so-called *golden section search*. In this approach, the reduction rate of the intervals of uncertainty for each iteration is the same, that is

$$x_1 - a_k = b_k - x_2 = \tau(b_k - a_k).$$

Substituting τ into (4.1) and solvinging yield

$$\tau = \frac{3 - \sqrt{5}}{2} \approx 0.382.$$

This rule was referred to by ancient Greek geometers as the Golden Section. Using this Golden Section rule means that at every stage of the uncertainty range reduction (except the first one), the objective function f need only be evaluated at one new point. The uncertainty range is reduced by the ratio $1 - \tau \approx 0.618$ at every stage. Hence, N steps of reduction using the Golden Section method reduces the range by the factor

$$(1 - \tau)^N \approx (0.618)^N.$$

The corresponding algorithm is stated as follows.

Example 4.3. Use the Golden Section search to find the value of x that minimizes

$$f(x) = x^4 - 14x^3 + 60x^2 - 70x$$

begin
Initialize initial interval $[a_0, b_0]$ *and tolerance* $\delta > 0$
Initialize the distinguishability constant $\varepsilon > 0$
$\tau :\approx 0.382$
while $b_k - a_k > \delta$
$x_1 := a_k + \tau(b_k - a_k)$
$x_2 := b_k - \tau(b_k - a_k)$
if $f(x_1) < f(x_2)$ **then**
$a_{k+1} := a_k$
$b_{k+1} := x_2$
else
$a_{k+1} := x_1$
$b_{k+1} := b_k$
end
end
$x^* := (a_k + b_k)/2$
end

Figure 4.4: The Golden Section Search Algorithm

in the range $[0, 2]$. Locate this value of x to within a range of 0.3.

After N stages the range $[0, 2]$ is reduced by $(0.618)^N$. So, we choose N so that

$$(0.618)^N < 0.3/2.$$

Four stages of reduction will do; that is, $N = 4$.

The first two observations are located at $x_1 = 0.764$, $x_2 = 1.236$. Note that $f(x_1) < f(x_2)$. Hence, the new interval of uncertainty is $[0.000, 1.236]$. The process is repeated, and the computations are summarized in Table 4.3. After four iterations involving ten observations, the interval of uncertainty is $[0.652, 0.944]$, so that the minimum can be estimated to be the midpoint 0.798. Note that the true minimum is in fact 0.781.

Iteration k	a_k	b_k	x_1	x_2	$f(x_1)$	$f(x_2)$
1	0.000	2.000	0.764	1.236	−24.361	−18.960
2	0.000	1.236	0.472	0.764	−21.099	−24.361
3	0.472	1.236	0.764	0.944	−24.361	−23.593
4	0.472	0.944	0.652	0.764	−23.838	−24.361

Table 4.3: Summary of computations for the Golden Section Search method

4.4. Newton's method

Suppose again that we are confronted with the problem of minimizing a function f of a single real variable x. We assume now that at each measurement point $x^{(k)}$ we can calculate $f(x^{(k)})$, $f'(x^{(k)})$, and $f''(x^{(k)})$. We can fit a quadratic function through $x^{(k)}$ that matches its first and second derivatives with that of the function f. This quadratic approximation F is given by

$$F(x) \triangleq f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)}) + \frac{1}{2} f''(x^{(k)})(x - x^{(k)})^2.$$

Then, instead of minimizing f, we minimize its approximation F. The point $x^{(k+1)}$ is taken to be the minimizer of F. The first-order necessary condition for a minimizer of F yields $F'(x) = f'(x^{(k)}) + f''(x^{(k)})(x - x^{(k)}) = 0$, so that

$$x^{(k+1)} := x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})} \tag{4.2}$$

The procedure is terminated when $|x^{(k+1)} - x^{(k)}| < \varepsilon$ or when $|f'(x^{(k)})| < \varepsilon$ where ε is a prespecified termination scalar. Note that the above procedure can only be applied for twice differentiable functions. Furthermore, the procedure is well defined only if $f''(x^{(k)}) \neq 0$ for each k.

The method of Newton, in general, does not converge to a stationary point starting with an arbitrary initial point. However, as shown in Theorem 4.2, if the starting point is sufficiently close to a stationary point, then a suitable descent function can be devised so that the method converges.

Theorem 4.2. Let $f(x)$ be continuously twice differentiable. Consider Newton's algorithm defined by (4.2). Let x^* be the stationary point of f and $f''(x^*) \neq 0$. Let the starting point $x^{(0)}$ be sufficiently close to x^* so that there exist positive scalars K_1 and K_2 with $K_1 K_2 < 1$ such that

1. $K_1 |f''(x)| \geq 1$;
2. $|f'(x^*) - f'(x) - f''(x)(x^* - x)| \leq K_2 |x^* - x|$

for each x satisfying $|x - x^*| \leq |x^{(0)} - x^*|$. Then the algorithm converges to x^*.

Exercises

1. The 5th-order polynomial

$$f(x) = -5x^5 + 4x^4 - 12x^3 + 11x^2 - 2x + 1$$

is known to be a unimodal function on interval $[-0.5, 0.5]$.

(a) Use the dichotomous search to find the minimizer of $f(x)$ on $[-0.5, 0.5]$ with the range of uncertainty less than 10^{-5}.

(b) Solve the line search problem in part (a) using the Fibonacci search.

(c) Solve the line search problem in part (a) using the golden-section search.

(d) Compare the computational efficiency of the methods in (a) – (c) in terms of number of function evaluations.

2. The function

$$f(x) = \log^2(x-2) + \log^2(10-x) - x^{0.2}$$

is known to be a unimodal function on $[6, 9.9]$.

(a) Use the dichotomous search to find the minimizer of $f(x)$ on $[6, 9.9]$ with the range of uncertainty less than 10^{-5}.

(b) Solve the line search problem in part (a) using the Fibonacci search.

(c) Solve the line search problem in part (a) using the golden-section search.

(d) Compare the computational efficiency of the methods in (a) – (c) in terms of number of function evaluations.

3. The function

$$f(x) = -3x \sin 0.75x + e^{-2x}$$

is known to be a unimodal function on $[0, 2\pi]$.

(a) Use the dichotomous search to find the minimizer of $f(x)$ on $[0, 2\pi]$ with the range of uncertainty less than 10^{-5}.

(b) Solve the line search problem in part (a) using the Fibonacci search.

(c) Solve the line search problem in part (a) using the golden-section search.

(d) Compare the computational efficiency of the methods in (a) – (c) in terms of number of function evaluations.

4. The function

$$f(x) = e^{3x} + 5e^{-2x}$$

is known to be a unimodal function on $[0, 1]$.

(a) Use the dichotomous search to find the minimizer of $f(x)$ on $[0, 1]$ with the range of uncertainty less than 10^{-5}.

(b) Solve the line search problem in part (a) using the Fibonacci search.

(c) Solve the line search problem in part (a) using the golden-section search.

(d) Compare the computational efficiency of the methods in (a) – (c) in terms of number of function evaluations.

5. The function

$$f(x) = 0.2x \log x + (x-2.3)^2$$

is known to be a unimodal function on $[0.5, 2.5]$.

(a) Use the dichotomous search to find the minimizer of $f(x)$ on $[0.5, 2.5]$ with the range of uncertainty less than 10^{-5}.

(b) Solve the line search problem in part (a) using the Fibonacci search.

(c) Solve the line search problem in part (a) using the golden-section search.

(d) Compare the computational efficiency of the methods in (a) – (c) in terms of number of function evaluations.

6. Let

$$f(x) = \sin^6 x \tan(1-x) e^{30x}.$$

Find the maximum of $f(x)$ in $[0, 1]$ by use of the golden-section search.

7. Let $\varphi(t) = e^{-t} + e^{t}$. Let the initial interval be $[-1, 1]$.

(a) Minimize $\varphi(t)$ by Fibonacci method.

(b) Solve the line search problem in part (a) using the Fibonacci search.

(c) Minimize $\varphi(t)$ by the golden-section search.

5. Basic multidimensional methods

In previous lectures, several methods were considered that can be used for the solution of one-dimensional unconstrained problems. Now we consider the solution of multidimensional unconstrained problems.

As for one-dimensional optimization, there are two general classes of multidimensional methods, namely, pattern search methods and gradient methods. In pattern search methods, the solution is obtained by using only function evaluations. The general approach is to explore the parameter space in an organized manner in order to find a trajectory that leads progressively to reduced values of the objective function. A rudimentary method of this class might be to adjust all the parameters at a specific starting point, one at a time, and then select a new point by comparing the calculated values of the objective function. The same procedure can then be repeated at the new point, and so on. Multidimensional search methods are thus analogous to their one-dimensional counterparts, and like the latter, they are not very efficient. As a result, their application is restricted to problems where gradient information is unavailable or difficult to obtain, for example, in applications where the objective function is not continuous.

Gradient methods are based on gradient information. They can be grouped into two subclasses, first-order and second-order methods. First-order methods are based on the linear approximation of the Taylor series, and hence they entail the gradient $\nabla f(\boldsymbol{x})$. Second-order methods, on the other hand, are based on the quadratic approximation of the Taylor series. They entail the gradient $\nabla f(\boldsymbol{x})$ as well as the Hessian $H(\boldsymbol{x})$.

5.1. Steepest-Descent Method

Assume that a function $f(\boldsymbol{x})$ is continuous in the neighborhood of point $\boldsymbol{x}$. A vector $\boldsymbol{d}$ satisfying

$$\boldsymbol{d}^{\mathrm{T}}\nabla f(\boldsymbol{x}) < 0$$

is called a *descent direction* of a function $f(\boldsymbol{x})$ at point $\boldsymbol{x}$. Since the value $\boldsymbol{d}^{\mathrm{T}}\nabla f(\boldsymbol{x})$ is the smallest if and only if $\boldsymbol{d} = -\nabla f(\boldsymbol{x})$, the direction $-\nabla f(\boldsymbol{x})$ is the *steepest descent direction* and it is a good direction to search if we want to find a function minimizer. If $\boldsymbol{d} \neq \mathbf{0}$ is the steepest-descent direction at point $\boldsymbol{x}$, then for sufficiently small $h > 0$, we have

$$f(\boldsymbol{x} + h\,\boldsymbol{d}) < f(\boldsymbol{x}).$$

This means that the point $\boldsymbol{x} + h\,\boldsymbol{d}$ is an improvement over the point $\boldsymbol{x}$ if we are searching for a minimizer. Maximum reduction in $f(\boldsymbol{x})$ can be achieved by solving the one-dimensional optimization problem

$$\min_{h \geq 0} f(\boldsymbol{x} + h\,\boldsymbol{d}). \tag{5.1}$$

If the steepest-descent direction at point $\boldsymbol{x}$ happens to point towards the minimum $\boldsymbol{x}^*$ of $f(\boldsymbol{x})$, then a value of h exists that minimizes $f(\boldsymbol{x} + h\,\boldsymbol{d})$ with respect to h and $f(\boldsymbol{x})$ with respect to $\boldsymbol{x}$. Consequently, in such a case the multidimensional problem can be solved by solving the one-dimensional problem in Eq. (5.1) once. Unfortunately, in most real-life problems, $\boldsymbol{d}$ does not point in the direction of $\boldsymbol{x}^*$ and, therefore, an iterative procedure must be used for the solution. Starting with an initial point $\boldsymbol{x}^{(0)}$, a direction $\boldsymbol{d} = \boldsymbol{d}^{(0)} = -\nabla f(\boldsymbol{x}^{(0)})$ can be calculated, and the value of h_0 that minimizes $f(\boldsymbol{x}^{(0)} + h\,\boldsymbol{d}^{(0)})$ can be determined. Thus a point $\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + h_0\,\boldsymbol{d}^{(0)}$ is obtained. The minimization can be performed by using one of the methods of the previous lecture as a line search. The same procedure can then be repeated at points

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + h_k\,\boldsymbol{d}^{(k)}$$

for $k = 1, 2, \ldots$ until convergence is achieved. The procedure can be terminated when $\|h_k\,\boldsymbol{d}^{(k)}\|$ becomes insignificant or if $h_k \leq K h_0$ where K is a sufficiently small positive constant. A corresponding algorithm is as follows.

---

begin
    Objective function $f(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$
    Initialize starting point $\boldsymbol{x}^{(0)}$ *and tolerance* $\varepsilon > 0$
    $k := 0$
    repeat
        Calculate gradient $\nabla f(\boldsymbol{x}^{(k)})$ *and set* $\boldsymbol{d}^{(k)} = -\nabla f(\boldsymbol{x}^{(k)})$
        Find h_k *such that* $f(\boldsymbol{x}^{(k)} + h_k\,\boldsymbol{d}^{(k)}) = \min_{h \geq 0} f(\boldsymbol{x}^{(k)} + h\,\boldsymbol{d}^{(k)})$
        $\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + h_k\,\boldsymbol{d}^{(k)}$
        $k := k + 1$
    while $\|h_k\,\boldsymbol{d}^{(k)}\| > \varepsilon$
    $\boldsymbol{x}^* := \boldsymbol{x}^{(k+1)}$
end

Figure 5.1: Steepest-descent algorithm

The steepest-descent method usually works quite well during early stages of the optimization process, depending on the point of initialization. However, as a stationary point is approached, the method usually behaves poorly, taking small orthogonal steps and therefore the trajectory to the solution follows a zigzag pattern. To demonstrate this fact, we note that

$$\frac{d}{dh} f(\boldsymbol{x}^{(k)} + h\,\boldsymbol{d}^{(k)}) = (\boldsymbol{d}^{(k)})^{\mathrm{T}} \nabla f(\boldsymbol{x}^{(k)} + h\,\boldsymbol{d}^{(k)})$$

If h^* is the value of h that minimizes $f(f(\boldsymbol{x}^{(k)} + h\,\boldsymbol{d}^{(k)})$, then

$$(\boldsymbol{d}^{(k)})^{\mathrm{T}} \nabla f(\boldsymbol{x}^{(k)} + h^*\,\boldsymbol{d}^{(k)}) = (\boldsymbol{d}^{(k)})^{\mathrm{T}} \boldsymbol{d}^{(k+1)} = 0.$$

In effect, successive directions $\boldsymbol{d}^{(k)}$ and $\boldsymbol{d}^{(k+1)}$ are orthogonal.

For continuously twice differentiable nonquadratic functions $f(\boldsymbol{x})$ that has a local minimizer at point $\boldsymbol{x}^*$ with positive definite Hessian, it can be shown that if $\boldsymbol{x}^{(k)}$ is sufficiently close to $\boldsymbol{x}^*$, we have

$$f(\boldsymbol{x}^{(k+1)}) - f(\boldsymbol{x}^*) \leq \left(\frac{1-r}{1+r}\right)^2 \left[f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^*)\right]$$

where

$$r = \frac{\text{smallest eigenvalue of } H(\boldsymbol{x}^{(k)})}{\text{largest eigenvalue of } H(\boldsymbol{x}^{(k)})}$$

Furthermore, if $f(\boldsymbol{x})$ is a quadratic function then the inequality holds for all k. In effect, subject to the conditions stated, the steepest-descent method converges linearly with a convergence ratio

$$\beta = \left(\frac{1-r}{1+r}\right)^2.$$

Evidently, the rate of convergence is high if the eigenvalues of $H(\boldsymbol{x}^{(k)})$ are all nearly equal, or low if at least one eigenvalue is small relative to the largest eigenvalue.

5.2. Newton's method

The steepest descent method is a first-order method based on the linear approximation of the Taylor series. It means that in selecting a suitable search direction method uses only first derivatives (gradients). This strategy is not always the most effective. If higher derivatives are used, the resulting iterative algorithm may perform better than the steepest descent method. Newton's method uses first and second derivatives and indeed does perform better than the steepest descent method if the initial point is close to the minimizer. The idea behind this method is as follows. Given a starting point, we construct a quadratic approximation to the objective function that matches the first and second derivative values at that point. We then minimize the approximate (quadratic) function instead of the original objective function. We use the minimizer of the approximate function as the starting point in the next step and repeat the procedure iteratively. If the objective function is quadratic, then the approximation is exact, and the method yields the true minimizer in one step. If, on the other hand, the objective function is not quadratic, then the approximation will provide only an estimate of the position of the true minimizer. Figure 5.2 illustrates the above idea.

We can obtain a quadratic approximation to the given twice continuously differentiable objection function f using the Taylor series expansion of f about the current point $\boldsymbol{x}^{(k)}$, neglecting terms of order three and higher. We obtain

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}^{(k)}) + (\boldsymbol{x} - \boldsymbol{x}^{(k)})^{\mathrm{T}} \nabla f(\boldsymbol{x}^{(k)}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{(k)})^{\mathrm{T}} H(\boldsymbol{x}^{(k)})(\boldsymbol{x} - \boldsymbol{x}^{(k)}) \triangleq F(\boldsymbol{x}).$$

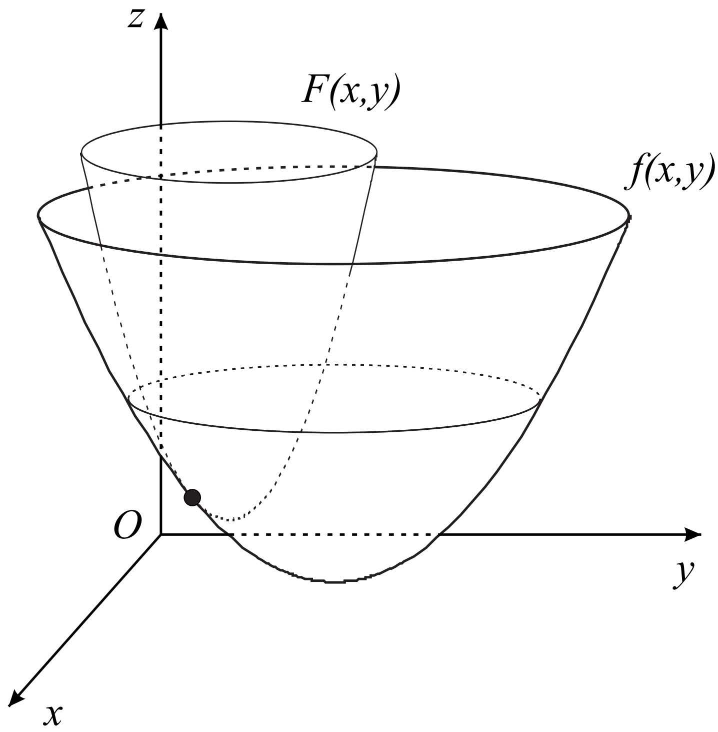

Figure 5.2: Quadratic approximation to the objective function using first and second derivatives

Applying the first-order necessary conditions to F yields

$$\mathbf{0} = \nabla F(\boldsymbol{x}) = \nabla f(\boldsymbol{x}^{(k)}) + H(\boldsymbol{x}^{(k)})(\boldsymbol{x} - \boldsymbol{x}^{(k)}).$$

If $H(\boldsymbol{x}^{(k)})$ is positive definite, then F achieves a minimum at

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{d}^{(k)}, \quad \boldsymbol{d}^{(k)} = -H^{-1}(\boldsymbol{x}^{(k)})\nabla f(\boldsymbol{x}^{(k)}).$$

This recursive formula represents Newton's method. The corresponding algorithm is stated as follows.

As in the one-variable case, there is no guarantee that Newton's algorithm heads in the direction of decreasing values of the objective function if $H(\boldsymbol{x}^{(k)})$ is not positive definite. Moreover, even if $H(\boldsymbol{x}^{(k)})$ is positive definite, Newton's method may not be a descent method; that is, it is possible that $f(\boldsymbol{x}^{(k+1)}) \geq f(\boldsymbol{x}^{(k)})$. For example, this may occur if our starting point $\boldsymbol{x}^{(0)}$ is far away from the solution. Despite the above drawbacks, Newton's method has superior convergence properties when the starting point is near the solution.

The convergence analysis of Newton's method when f is a quadratic function is straightforward. In fact, Newton's method reaches the point $\boldsymbol{x}^*$ such that $\nabla f(\boldsymbol{x}^*) = 0$ in just one step starting from any initial point $\boldsymbol{x}^{(0)}$. To see this, suppose that $Q = Q^{\mathrm{T}}$ is invertible, and

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{b}, \quad \nabla f(\boldsymbol{x}) = Q\boldsymbol{x} - \boldsymbol{b}, \quad \text{and} \quad H(\boldsymbol{x}) = Q.$$

begin
Objective function $f(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$
Initialize starting point $\boldsymbol{x}^{(0)}$ *and tolerance* $\varepsilon > 0$
$k := 0$
while $\|\nabla f(\boldsymbol{x}^{(k)})\| > \varepsilon$
Solve $H(\boldsymbol{x}^{(k)})\boldsymbol{d}^{(k)} = -\nabla f(\boldsymbol{x}^{(k)})$ *for* $\boldsymbol{d}^{(k)}$
$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \boldsymbol{d}^{(k)}$
$k := k + 1$
end
$\boldsymbol{x}^* := \boldsymbol{x}^{(k)}$
end

Figure 5.3: Newton's method algorithm

Hence, given any initial point $\boldsymbol{x}^{(0)}$ by Newton's algorithm

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} - H^{-1}(\boldsymbol{x}^{(0)})\nabla f(\boldsymbol{x}^{(0)}) = \boldsymbol{x}^{(0)} - Q^{-1}\left[Q\boldsymbol{x}^{(0)} - \boldsymbol{b}\right] = Q^{-1}\boldsymbol{b} = \boldsymbol{x}^*.$$

Therefore, for any initial point $\boldsymbol{x}^{(0)}$ one iteration will yield the solution.

To analyze the convergence of Newton's method in the general case, we formulate the following theorem.

Theorem 5.1. Suppose that $f \in C^3$, and $\boldsymbol{x}^* \in \mathbb{R}^n$ is a point such that $\nabla f(\boldsymbol{x}^*) = 0$ and $H(\boldsymbol{x}^*)$ is invertible. Then, for all $\boldsymbol{x}^{(0)}$ sufficiently close to $\boldsymbol{x}^*$, Newton 's method is well defined for all k, and converges to $\boldsymbol{x}^*$ with order of convergence at least 2.

As stated in the above theorem, Newton's method has superior convergence properties if the starting point is near the solution. However, the method is not guaranteed to converge to the solution if we start far away from it. In particular, the method may not be a descent method. Fortunately, it is possible to modify the algorithm such that the descent property holds:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - h_k\, \boldsymbol{d}^{(k)}$$

where

$$h_k = \arg\min_{h \geq 0} f(\boldsymbol{x}^{(k)} - h\, \boldsymbol{d}^{(k)}),$$

that is, at each iteration, we perform a line search in the direction $-\boldsymbol{d}^{(k)}$. So, we conclude that the above modified Newton's method has the descent property; that is,

$$f(\boldsymbol{x}^{(k+1)}) < f(\boldsymbol{x}^{(k)})$$

whenever $\nabla f(\boldsymbol{x}^{(k)}) \neq 0$. The corresponding algorithm is stated as follows.

begin
Objective function $f(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$
Initialize starting point $\boldsymbol{x}^{(0)}$ *and tolerance* $\varepsilon > 0$
$k := 0$
while $\|\nabla f(\boldsymbol{x}^{(k)})\| > \varepsilon$
Solve $H(\boldsymbol{x}^{(k)})\boldsymbol{d}^{(k)} = -\nabla f(\boldsymbol{x}^{(k)})$ *for* $\boldsymbol{d}^{(k)}$
Find h_k *such that* $f(\boldsymbol{x}^{(k)} + h_k\, \boldsymbol{d}^{(k)}) = \min_{h \geq 0} f(\boldsymbol{x}^{(k)} + h\, \boldsymbol{d}^{(k)})$
$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + h_k\, \boldsymbol{d}^{(k)}$
$k := k + 1$
end
$\boldsymbol{x}^* := \boldsymbol{x}^{(k)}$
end

Figure 5.4: Newton's method with line search algorithm

Another source of potential problems in Newton's method arises from the Hessian matrix not being positive definite. A simple technique to ensure that the search direction is a descent direction is to introduce the so-called *Levenberg-Marquardt* modification to Newton's algorithm:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \left[H^{-1}(\boldsymbol{x}^{(k)}) + \mu_k I\right]^{-1} \nabla f(\boldsymbol{x}^{(k)}),$$

where $\mu_k \geq 0$.

The idea underlying the Levenberg-Marquardt modification is as follows. Consider a symmetric matrix F, which may not be positive definite. Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of F with corresponding eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$. The eigenvalues $\lambda_1, \ldots, \lambda_n$ are real, but may not all be positive. Next, consider the matrix $G = F + \mu U$, where $\mu \geq 0$. Note that the eigenvalues of G are $\lambda_1 + \mu, \ldots, \lambda_n + \mu$. Indeed,

$$G\boldsymbol{v}_i = (F + \mu I)\boldsymbol{v}_i = F\boldsymbol{v}_i + \mu I\boldsymbol{v}_i = \lambda_i\boldsymbol{v}_i + \mu\boldsymbol{v}_i = (\lambda_i + \mu)\boldsymbol{v}_i,$$

which shows that for all $i = 1, \ldots, n$, $\boldsymbol{v}_i$ is also an eigenvector of G with eigenvalue $\lambda_i + \mu$. Therefore, if μ is sufficiently large, then all the eigenvalues of G are positive, and G is positive definite. Accordingly, if the parameter μ_k in the Levenberg-Marquardt modification of Newton's algorithm is sufficiently large, then the search direction $\boldsymbol{d}^{(k)} = -\left[H^{-1}(\boldsymbol{x}^{(k)}) + \mu_k I\right]^{-1} \nabla f(\boldsymbol{x}^{(k)})$ always points in a descent direction. In this case, if we further introduce a step size h_k as described in the previous section,

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - h_k \left[H^{-1}(\boldsymbol{x}^{(k)}) + \mu_k I\right]^{-1} \nabla f(\boldsymbol{x}^{(k)}),$$

then we are guaranteed that the descent property holds.

The Levenberg-Marquardt modification of Newton's algorithm can be made to approach the behavior of the pure Newton's method by letting $\mu_k \to 0$. On the other hand, by letting $\mu_k \to \infty$, the algorithm approaches a pure gradient method with small step size. In practice, we may start with a small value of μ_k, and then slowly increase it until we find that the iteration is descent, that is, $f(\boldsymbol{x}^{(k+1)}) < f(\boldsymbol{x}^{(k)})$.

5.3. Hooke-Jeeves pattern search

As we said above, the pattern search method works by creating a set of search directions iteratively. The created search directions should be such that they completely span the search space. In other words, they should be such that starting from any point in the search space any other point in the search space can be reached by traversing along these search directions only. In a n-dimensional problem, this requires at least n linearly independent search directions. For example, in a two-variable function, at least two search directions are required to go from any one point to any other point. Among many possible combinations of n search directions, some combinations maybe able to reach the destination faster (with lesser iterations), and some may require more iterations.

The Hooke-Jeeves pattern search method is one of such methods consisted of two moves: exploratory move and heuristic pattern move. The exploratory moves explore the local behaviour and information of the objective function so as to identify any potential sloping valleys if they exist. For any given step size Δ, exploration movement performs from an initial starting point along each coordinate direction by increasing or decreasing $\pm\Delta$, if the new value of the objective function does not increase (for minimization problem), that is $f(\boldsymbol{x}^{(k)}+\Delta\, \boldsymbol{e}_i) < f(\boldsymbol{x}^{(k)})$, the exploratory move is considered as successful. If it is not successful, then try a step in the opposite direction, and the result is updated only if it is successful. When all the n coordinates have been explored, the resulting point forms a base point $\boldsymbol{x}^{(k)}$.

The pattern move intends to move the current base $\boldsymbol{x}^{(k)}$ along the base line $\boldsymbol{x}^{(k)}-$ $-\boldsymbol{x}^{(k-1)}$ from the previous (historical) base point to the current base point. The move is carried out by the following formula

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + 2\left[\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\right].$$

Then $\boldsymbol{x}^{(k+1)}$ forms a new temporary base point for further new exploratory moves. If the pattern move produces improvement (lower value of $f(\boldsymbol{x})$), the new base point $\boldsymbol{x}^{(k+1)}$ is successfully updated. If the pattern move does not lead to improvement or lower value of the objective function, then the pattern move is discarded and new search starts from $\boldsymbol{x}^{(k)}$, and new search moves should use smaller step size $\Delta/2$. Iterations continue until the prescribed tolerance ε is met. The algorithm is summarised in the pseudo code shown in Figure 5.5.

Exercises

1. Consider the minimization problem

$$\min\{x_1^2 + x_2^2 - 0.2x_1x_2 - 2.2x_1 + 2.2x_2 + 2.2\}$$

(a) Find a point satisfying the First-Order Necessary Conditions for a minimizer.

(b) Show that this point is the global minimizer.

(c) What is the rate of convergence of the steepest-descent method for this problem?

(d) Starting at $\boldsymbol{x}^{(0} = (1, 1)^{\mathrm{T}}$, how many steepest-descent iterations would it take (at most) to reduce the function value to 10^{-10}?

2. Solve the problem

$$\min\{5x_1^2 - 9x_1x_2 + 4.075x_2^2 + x_1\}$$

by applying the steepest-descent method with $\boldsymbol{x}^{(0} = (1, 1)^{\mathrm{T}}$ and $\varepsilon = 3 \times 10^{-6}$. Give a convergence analysis on the above problem to explain why the steepest-decent method requires a large number of iterations to reach the solution.

3. Solve the problem

$$\min\{2x_1^2 - 2x_1x_2 + x_2^2 + 2x_1 - 2x_2\}$$

with $\boldsymbol{x}^{(0} = (1, 1)^{\mathrm{T}}$ by using Newton method.

4. Solve the problem

$$\min\{x_1^2 + 2x_2^2 + 4x_1 + 4x_2\}$$

with $\boldsymbol{x}^{(0} = (1, 1)^{\mathrm{T}}$ by using Newton method.

5. Solve the problem

$$\min\{2x_1^2 - 2x_1x_2 + x_2^2 + 2x_1 - 2x_2\}$$

with $\boldsymbol{x}^{(0} = (1, 1)^{\mathrm{T}}$ by using Hooke-Jeeves method.

6. Solve the problem

$$\min\{x_1^2 + 2x_2^2 + 4x_1 + 4x_2\}$$

with $\boldsymbol{x}^{(0} = (1, 1)^{\mathrm{T}}$ by using Hooke-Jeeves method.

```
begin
    Objective function f(x), x = (x_1, ..., x_n)^T
    Let e_1, ..., e_n be the coordinate directions
    Initialize tolerance ε > 0 and step size Δ ≥ ε
    Initialize starting point x^(0)
    y := x^(0)
    k := 0
    while Δ > ε
        loop
            for i := 1 to n do
                if f(y + Δ e_i) < f(y) then
                    y := y + Δ e_i
                else if f(y − Δ e_i) < f(y) then
                    y := y − Δ e_i
                end
            end
            if f(y) ≥ f(x^(k)) then
                return
            end
            x^(k+1) := y
            y := x^(k+1) + 2 [x^(k+1) − x^(k)]
            k := k + 1
        end loop
        Δ := Δ/2
        y := x^(k)
        x^(k+1) := x^(k)
        k := k + 1
    end
end
```

Figure 5.5: Hooke-Jeeves pattern search algorithm

Bibliography

[1] Yang - Introduction to mathematical optimization

[2] A. Antoniou and W.S. Lu, Practical optimization. Algorithms and engineer-ing applications. Springer, New York, 2007.

[3] M. Bartholomew-Biggs, Nonlinear optimization with engineering applications. Springer, New York, 2008.

[4] M.S. Bazaraa, H.D. Sherali and C.M. Shetty, Nonlinear Programming: Theory and Algorithms. 3rd ed., Wiley, 2006.

[5] B. Kolman, R.E. Beck, Elementary Linear Programming with Applications, Gulf Professional Publishing, 1995.

[6] H.A. Eiselt and C.L. Sandblom, Linear Programming and its Applications. Springer, New York, 2007.

[7] I.M. Gelfand and S.V. Fomin, Calculus of variations. Prentice-Hall, 1963.

[8] J. Schneider and S. Kirkpatrick, Stochastic optimization. Springer, New York, 2006.

[9] A. Shapiro, D. Dentcheva and A. Ruszczynski, Lectures on Stochastic Programming: Modeling and Theory. SIAM, 2009.

[10] D. Bertsekas, Nonlinear programming. 2nd ed., Athena, 1999.

[11] E.K.P. Chong and S.H. Zak, An introduction to optimization. 2nd ed., Wiley, 2001.

[12] G.B. Dantzig and M.N. Thapa, Linear Programming 2: Theory and Exten-sions. Springer-Verlag, New York, 2003.

[13] U. Diwekar, Introduction to Applied Optimization. Series: Springer Optimization and Its Applications, Vol. 22, 2nd ed., 2008.

[14] A.V. Fiacco and G.P. McCormick, Nonlinear programming. SIAM, 1990.

[15] G. Giorgi, A. Guerraggio and J. Thierfelder, Mathematics of optimization. Elsevier, 2004.

[16] J. Pinter, Global optimization in action: Continuous and Lipschitz optimization. Kluwer, 1996.

A. Course Content

1. Introduction
   Historical development. Engineering application of optimization. Formulation of design problems as mathematical programming problems. Classification of optimization problems.

2. Unconstrained Optimization
   The Gradient, Hessian and Jacobian. The Taylor Series. Symmetric and positive definite matrices. Types of extrema. Necessary and sufficient conditions for local minima and maxima. Classification of stationary points.

3. Linear Programming

   Geometry of linear programming
   Standard Form. Basic Solutions and Extreme Points. Representation of Solutions. Optimality. Two-dimensional linear programs. The Graphical Solution Technique.

   Simplex method
   The simplex method. Unbounded problems. Notation for the simplex method. Deficiencies of the tableau. Multiple solutions. Feasible directions. Edge directions. Artificial variables. Degeneracy and termination. The two-phase method. Resolving degeneracy using perturbation.

   Duality and sensitivity
   Formulation of the dual problem. Primal-dual relationships. The dual simplex method. The Primal-Dual Method. Finding an Initial Dual Feasible Solution.

   Applications
   The Diet Problem. Allocation Problems. Cutting Stock Problems. Employee Scheduling. Data Envelopment Analysis. Inventory Planning. Blending Problems. Transportation Problems. Assignment Problems.

4. Nonlinear Constrained Optimization

   Problems with equality constraints Problem formulation. Optimality conditions for linear equality constraints. The Lagrange multipliers and the Lagrangian function. Lagrange condition. Second-order conditions.

   Problems with inequality constraints Karush-Kuhn-Tucker condition. Second-order conditions.

5. Numerical Optimization Algorithms

   One-dimensional search methods
   Unimodal function. Definition and properties. Dichotomous Search. Ternary

search. Golden section search. Fibonacci search. Newton's method. Secant method.

Basic multidimensional gradient methods

The method of steepest descent. Analysis of gradient methods. Convergence. Newton's method. Analysis of Newton's method. Levenberg-Marquardt modification. Convergence. Gradient-free methods: Hooke-Jeeves pattern search

Global optimization algorithms

Lipschitz condition. Properties. Piyavskii's method. Ternary search. Convergence.

B. Samples of Exam Tasks

Problem 1. Please answer true or false. No explanation is needed.

1. A problem of maximizing a convex, piecewise-linear function over a polyhedron can be formulated as a linear programming problem.
2. The dual of the problem min x_1 subject to $x_1 = 0, x_1, x_2 \geq 0$ has a nondegenerate optimal solution.
3. If there is a nondegenerate optimal basis, then there is a unique optimal basis.
4. An optimal basic feasible solution is strictly complementary.
5. If the primal linear programming problem has multiple optimal solutions, then there is at least one degenerate basic feasible primal solution.
6. Let $\theta \in \mathbb{R}$. The function $F(\theta) := \{\max \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}, x_i \leq \theta\}$ is concave.
7. The only way that the simplex method can get to a degenerate basic feasible solutionat some iteration is when there is a tie in the min-ratio test in the previous iteration.
8. The dual simplex method can detect whether a linear optimization problem is infeasible or unbounded.
9. The column geometry shows that the number of simplex pivots to find anoptimal basic feasible solution is linear in the dimension n.
10. Given a local optimum $\boldsymbol{x}^*$ for a nonlinear optimization problem it always satisfies the Kuhn-Tucker conditions when the gradients of the tight constraints and the gradients of the equality constraints at the point $\boldsymbol{x}^*$ are linearly independent.
11. The convergence of the steepest descent method for quadratic problems min $\boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x}$ highly depends on the condition number of the matrix Q. The larger the condition number the slower the convergence.
12. The convergence of the steepest descent method highly depends on the starting point.
13. For quadratic functions, Newton's method converges exactly in one iteration.
14. For a nonlinear optimization problem, if Newton's method converges, then it converges to a local minimum.
15. The larger the condition number of the Hessian, the slower the convergence of Newton's method.

Problem 2. Find solution of the following problems.

1. A steel company must decide how to allocate next week's time on a rolling mill, which is a machine that takes unfinished slabs of steel as input and can produce

either of two semi-finished products: bands and coils. The mill's two products come off the rolling line at different rates: Bands 200 tons/hr & Coils 140 tons/hr. They also produce different profits: Bands $ 25/ton & Coils $ 30/ton. Based on currently booked orders, the following upper bounds are placed on the amount of each product to produce: Bands 6000 tons & Coils 4000 tons. Given that there are 40 hours of production time available this week, the problem is to decide how many tons of bands and how many tons of coils should be produced to yield the greatest profit.

2. When the gates open at Sidney Planet amusement park, the patrons all rush to the Spaced Out Center. One path goes through Mikey Moose Square. The path from the entrance to the square can handle up to 500 patrons and the path from the square to the center can handle at most 400 patrons. Patrons can also go from the entrance via Gumbo's Restaurant and then Large World Park with the path on the first leg handling up to 300 patrons, the second leg only 100 patrons and the third leg at most 400 patrons. There are also paths from Gumbo's to the square and from the square to the park handling no more than 100 and 300 patrons, respectively.

3. Hal's Refinery can buy two types of gasoline. Boosch Oil has available, at $60 per barrel, 130, 000 barrels of 92 octane gasoline with vapor pressure 4.6 psi and sulfur content 0.58%. Chayni Oil has available, at $70 per barrel, 140, 000 barrels of 85 octane gasoline with vapor pressure 6.5 psi and sulfur content 0.40%. Hal needs to blend these two to produce at least 200, 000 barrels of a mixture with octane between 87 and 89, with vapor pressure at most 6.0 psi and sulfur content at most 0.50%.

Руслан Сергеевич **Бирюков**

МЕТОДЫ ОПТИМИЗАЦИИ

Учебно-методическое пособие

Государственное образовательное учреждение высшего
профессионального образования «Нижегородский государственный
университет им. Н.И. Лобачевского».
603950, Нижний Новгород, пр. Гагарина, 23.

Подписано в печать 29.11.2005. Формат 60x84 1/16.
Бумага офсетная. Печать офсетная. Гарнитура Таймс.
Усл. печ. л. 3,0. Уч.-изд. л. 3,3.
Заказ № 325. Тираж 100 экз.

Отпечатано в типографии Нижегородского госуниверситета
им. Н.И. Лобачевского
603600, г. Нижний Новгород, ул. Большая Покровская, 37
Лицензия ПД № 18-0099 от 14.05.01