The simplex algorithm

Lars Hesselholt

1. The optimization problem

We consider the following optimization problem. We wish to find the maximum value of the *linear* function in n variables

$$f(x_1, \dots, x_n) = c_1x_1 + \dots + c_nx_n \tag{1.1}$$

under the assumption that variables $x_1, \dots, x_n$ be non-negative and satisfy the m linear inequalities

$$\begin{array}{rcl} a_{1,1}x_{1,1} + a_{1,2}x_{1,2} + \dots + a_{1,n}x_{1,n} & \leqslant & b_1 \\ a_{2,1}x_{1,1} + a_{2,2}x_{1,2} + \dots + a_{2,n}x_{1,n} & \leqslant & b_2 \\ & \vdots & \\ a_{m,1}x_{1,1} + a_{m,2}x_{1,2} + \dots + a_{m,n}x_{1,n} & \leqslant & b_m \end{array} \tag{1.2}$$

Let $H_i \subset \mathbb{R}^n$ be the subset of solutions to the ith linear inequality in (1.2), and let $H_j' \subset \mathbb{R}^n$ be the set of solutions to the linear inequality $x_j \geqslant 0$. Then H_i and H_j' are closed halfspaces of $\mathbb{R}^n$. We wish to find the maximum value of the function (1.1) on the *feasible region* $P \subset \mathbb{R}^n$ defined by the intersection

$$P = \Big(\bigcap_{1 \leqslant i \leqslant n} H_i'\Big) \cap \Big(\bigcap_{1 \leqslant j \leqslant m} H_j\Big). \tag{1.3}$$

The structure of the feasible region P is given by the main theorem of polytopes which we now recall. We refer to [**1**, Thm. 1.1] for the proof.

THEOREM 1.4. *Let $P \subset \mathbb{R}^n$ be the intersection of a finite number of closed halfspaces and suppose that $P \subset \mathbb{R}^n$ is bounded. Then P is equal to the convex hull of a finite subset $V \subset \mathbb{R}^n$.*

We recall that the *convex hull* of the finite subset $V = \{v_1, \dots, v_N\} \subset \mathbb{R}^n$ is defined to be the subset $P \subset \mathbb{R}^n$ of all linear combinations $a_1v_1 + \dots + a_Nv_N$ such that all $a_i \geqslant 0$ and $a_1 + \dots + a_N = 1$. The convex hull $P \subset \mathbb{R}^n$ of a finite subset of $V \subset \mathbb{R}^n$ is called a *convex polytope*, and if V is minimal with this property, then V is called the set of *vertices* of the convex polytope P.

COROLLARY 1.5. *Let $P \subset \mathbb{R}^n$ be the convex hull of the finite set $V \subset P$, and let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a linear function. Then*

$$\max\{f(x) \mid x \in P\} = \max\{f(v) \mid v \in V\}.$$

Proof. Since $V \subset P$, we clearly have

$$\max\{f(x) \mid x \in P\} \geqslant \max\{f(v) \mid v \in V\}.$$

Conversely, every element $x \in P$ can be written $x = \sum_{v \in V} a_v v$ with $a_v \geqslant 0$ and $\sum_{v \in V} a_v = 1$. Since f is linear, we find

$$f(x) = \sum_{v \in V} a_v f(v) \leqslant \sum_{v \in V} a_v \max\{f(v) \mid v \in V\} = \max\{f(v) \mid v \in V\}.$$

This completes the proof. □

We return to the optimization problem at hand. Suppose that the feasible region P is bounded. Then Cor. 1.5 shows that to find the maximum of the function f on P, we need only compare the values of f at the finitely many vertices of P. However, in real world examples, the number of vertices will be too large for this to be practical.

2. A reformulation

Two vertices in the feasible region P are called *neighbors* if the line segment between them lies in the boundary of P. The simplex algorithm produces a sequence

$$v_0, v_1, \ldots, v_s \ldots \tag{2.1}$$

of vertices of the convex polytope P such that v_s and v_{s-1} are neighbors and $f(v_{s-1}) \leqslant f(v_s)$. The increasing sequence of values

$$f(v_0), f(v_1), \ldots, f(v_s), \ldots \tag{2.2}$$

becomes constant, for s large, and this constant value is the desired maximum value of the optimization problem. In short, the algorithm is as follows. We first choose some vertex v_0, say, the origin. Then, given v_{s-1}, we choose v_s to be a neighboring vertex of v_{s-1} for which the increase in the value of f is maximal.

It is possible to construct bad examples where the simplex algorithm visits all vertices of P before arriving at the optimal vertex. In practice, however, the algorithm is very effective. Moreover, it is easy to implement as we will now see.

We first rewrite the linear inequalities (1.2) and the requirement that the variables $x_1, \ldots, x_n$ be non-negative in the following equivalent form. We introduce m additional variables $x_{n+1}, \ldots, x_{m+n}$ and ask that the linear equations

$$\begin{array}{rcl} a_{1,1}x_{1,1} + a_{1,2}x_{1,2} + \cdots + a_{1,n}x_{1,n} + x_{n+1} & = & b_1 \\ a_{2,1}x_{1,1} + a_{2,2}x_{1,2} + \cdots + a_{2,n}x_{1,n} + x_{n+2} & = & b_2 \\ & \vdots & \\ a_{m,1}x_{1,1} + a_{m,2}x_{1,2} + \cdots + a_{m,n}x_{1,n} + x_{m+n} & = & b_m \end{array} \tag{2.3}$$

be satisfied and that all variables $x_1, \ldots, x_n, x_{n+1}, \ldots, x_{m+n}$ be non-negative. The new variables $x_{n+1}, \ldots, x_{m+n}$ are called the *slack variables* and measure how close the original inequalities are to being equalities. Written in matrix form, this of linear equation becomes

$$
\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} & 1 & 0 & \dots & 0 \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{m+n} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \tag{2.4}
$$

We say that a solution $(x_1, \dots, x_{m+n})$ to the system of linear equations (2.4) is a *feasible solution*, if all $x_1, \dots, x_{m+n}$ are non-negative, and we say that a feasible solution $(x_1, \dots, x_{m+n})$ is a *basic feasible solution*, if exactly n of the variables $x_1, \dots, x_{m+n}$ are equal to zero. The map

$$(x_1, \dots, x_n, x_{n+1}, \dots, x_{m+n}) \mapsto (x_1, \dots, x_n)$$

that forgets the slack variables gives a one-to-one correspondance between the set of feasible solutions (resp. basic feasible solutions) of (2.4) and the set of points (resp. vertices) of the feasible region (1.3). Moreover, two vertices in the fesible region (1.3) are neighbors if and only if the corresponding basic feasible solutions of (2.4) share exactly $n-1$ zeros.

3. The algorithm

We now explain Danzig's algorithm for solving the optimization problem. It is customary to write the system of linear equations (2.4) in the following form which is called a *tableau*. We also include, as the bottom row, the negative of the coefficients of the function $f(x_1, \dots, x_n)$ we wish to maximize.

x_1	x_2	$\dots$	x_n	x_{n+1}	x_{n+2}	$\dots$	x_{m+n}	
$a_{1,1}$	$a_{1,2}$	$\dots$	$a_{1,n}$	1	0	$\dots$	0	b_1
$a_{2,1}$	$a_{2,2}$	$\dots$	$a_{2,n}$	0	1	$\dots$	0	b_2
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
$a_{m,1}$	$a_{m,2}$	$\dots$	$a_{m,n}$	0	0	$\dots$	1	b_m
$-c_1$	$-c_2$		$-c_n$	0	0		0	

The tableau above is called the initial tableau. The algorithm also uses a *partition* of the variables $x_1, \dots, x_n, x_{n+1}, \dots, x_m$ into a group m variables called the *basic variables* and a group of n variables called the *non-basic variables*. In the initial partition, the m slack variables $x_{n+1}, \dots, x_{m+n}$ are the basic variables and the original variables $x_1, \dots, x_n$ are the non-basic variables. Each iteration of the algorithm produces a new tableau and a new partition of the variables.

Step 1: Are all the entries in the bottom row of the tableau non-negative? If yes, stop; the current tableau is the final tableau. If no, go to Step 2.

Step 2: Choose a variable x_s such that the corresponding entry in the bottom row has as large a negative value as possible. We call x_s the *entering basic variable* and we call the sth column the *pivot column*.

Step 3: Does there exist a positive entry in the pivot column between the two horizontal lines? If yes, go to Step 4. If no, stop; the function f is unbounded on the feasible region P (which is unbounded).

Step 4: Let $\alpha_{i,s}$ be the ith entry in the pivot column, and let β_i be the ith entry in the column on the right-hand side of the vertical line. Then among the positive entries in the pivot column, we choose an entry $\alpha_{r,s}$ such that $\beta_r/\alpha_{r,s}$ is as small as possible. The entry $\alpha_{r,s}$ is called the *pivot*. There is a unique basic variable x_t such that, in the tth column, the rth entry $\alpha_{r,t}$ is non-zero. We call x_t the *leaving basic variable*.

Step 5: Change the partition of the variables such that the entering basic variable x_s becomes a basic variable and the leaving basic variable x_t becomes a non-basic variable. Change the tableau by applying elementary row operations as follows. Let R_i be the ith row. (We include the bottom row.) Then, for $i \neq r$, we replace the ith row R_i by $\alpha_{r,s}R_i - \alpha_{i,s}R_r$.

Given the final tableau, we find the *final basic feasible solution* as follows. We set all non-basic variables in the final partition equal to zero and solve for the basic variables (using the final tableau). The final basic feasible solution is the solution to the optimization problem.

4. An example

Let us work out a simple example. We wish to maximize the function

$$f(x_1, x_2, x_3) = x_1 + 2x_2 + 3x_3$$

subject to the constraints that x_1, x_2, and x_3 be non-negative and satisfy the linear inequalities

$$\begin{aligned} 7x_1 + x_3 &\leqslant 6 \\ x_1 + 2x_2 &\leqslant 20 \\ 3x_2 + 4x_3 &\leqslant 30 \end{aligned}$$

Since we have 3 linear inequalities, we introduce 3 slack variables x_4, x_5, and x_6. The initial tableau takes the form

x_1	x_2	x_3	x_4	x_5	x_6	
7	0	1	1	0	0	6
1	2	0	0	1	0	20
0	3	4	0	0	1	30
-1	-2	-3	0	0	0	

and the initial basic variables are x_4, x_5, and x_6. We apply the simplex algorithm.

Iteration 1:

Step 1: Not all entries in the bottom row are non-negative.

Step 2: The entering basic variable is x_3 and the third column is the pivot column.

Step 3: There are positive entries in the pivot column.

Step 4: Since $30/4 > 6/1$, the pivot is the top entry 1 of the pivot column. The leaving basic variable is x_4.

Step 5: The new basic variables are x_3, x_5, and x_6. To find the new tableau we replace R_2 by $3R_2 - 2R_3$ and R_4 by $3R_4 + 2R_3$. It takes the form:

x_1	x_2	x_3	x_4	x_5	x_6	
7	0	1	1	0	0	6
1	2	0	0	1	0	20
−28	3	0	−4	0	1	6
20	−2	0	3	0	0	

Iteration 2:

Step 1: Not all entries in the bottom row are non-negative.

Step 2: The entering basic variable is x_2 and the second column is the pivot column.

Step 3: There are positive entries in the pivot column.

Step 4: Since $20/2 > 6/3$, the pivot is 3. The leaving basic variable is x_6.

Step 5: The new basic variables are x_2, x_3, and x_5. To find the new tableau, we replace R_2 by $3R_2 - 2R_3$ and R_4 by $3R_4 + 2R_3$. It takes the form

x_1	x_2	x_3	x_4	x_5	x_6	
7	0	1	1	0	0	6
59	0	0	8	3	−2	48
−28	3	0	−4	0	1	6
4	0	0	1	0	2	3

Iteration 3:

Step 1: All entries in the bottom row are non-negative.

We calculate the final basic feasible solution. We set the non-basic variables x_1, x_4, and x_6 equal to zero, and find $x_2 = 2$, $x_3 = 6$, and $x_5 = 16$. Hence, the desired maximal value of the function f is $f(0, 2, 6) = 22$.

References

[1] G. M. Ziegler, *Lectures on polytopes*, Graduate Texts in Mathematics, vol. 152, Springer-Verlag, New York, 1995.

Nagoya University, Nagoya, Japan

E-mail address: larsh@math.nagoya-u.ac.jp