ScienceDirect

Simplex Method

Related terms:

Linear Programming, Artificial Variable, Linear Programming Problem, Nonbasic Variable

View all Topics

An Introduction to Optimization

Richard Bronson, Gabriel B. Costa, in Matrix Methods (Third Edition), 2009

Problems 4.4

Using the Simplex Method, solve the following problems:

1. Section 4.2, Problem (1).
2. Section 4.2, Problem (2).
3. Maximize $z = 3x_1 + 5x_2$, subject to $x_1 + x_2 \leq 6$ and $2x_1 + x_2 \leq 8$.
4. Maximize $z = 8x_1 + x_2$, subject to the same constraints in (3).
5. Maximize $z = x_1 + 12x_2$, subject to the same constraints in (3).
6. Maximize $z = 3x_1 + 6x_2$, subject to the constraints $x_1 + 3x_2 \leq 30$, $2x_1 + 2x_2 \leq 40$, and $3x_1 + x_2 \leq 30$.
7. Consider problem (9) at the end of Section 4.2. Set up the initial tableaus for problems (9a) through (9d).

> Read full chapter

Linear Programming Methods for Optimum Design

Jasbir S. Arora, in Introduction to Optimum Design (Second Edition), 2004

6.3.4 The Pivot Step

In the Simplex method, we want to systematically search among the basic feasible solutions for the optimum design. *We must have a basic feasible solution to initiate the* Simplex method. Starting from the basic feasible solution, we want to find another that decreases the cost function. This can be done by interchanging a current basic variable with a nonbasic variable. That is, a current basic variable is made nonbasic (i.e., reduced to 0 from a positive value) and a current nonbasic variable is made basic (i.e., increased from 0 to a positive value). The *pivot step* accomplishes this task and results in a new canonical form (general solution), as explained in the following.

Let us select a basic variable xp ($1 \le p \le m$) to be replaced by a nonbasic variable xq for $(n - m) \le q \le n$. We will describe later how to determine xp and xq. The pth basic column is to be interchanged with the qth nonbasic column. This is possible only when the element in the pth column and qth row is nonzero; i.e., $apq \ne 0$. The element $apq \ne 0$ is called the *pivot element. The pivot element must always be positive* in the Simplex method as we shall see later. Note that xq will be basic if it is eliminated from all the equations except the pth one. This can be accomplished by performing a Gauss-Jordan elimination step on the qth column of the tableau shown in Table 6-3 using the pth row for elimination. This will give $apq = 1$ and zeros elsewhere in the qth column. The row used for the elimination process (pth row) is called the pivot row. The column on which the elimination is performed (qth column) is called the pivot column. The process of interchanging one basic variable with a nonbasic variable is called the *pivot step*.

Let a_{ij}' denote the new coefficients in the canonical form after the pivot step. Then, the *pivot step* for performing elimination in the qth column using the pth row as the pivot row is described by the following general equations.

Divide the pivot row (p) by the pivot element apq:

(6.15)

Eliminate xq from all rows except the pth row:

(6.16)

(6.17)

In Eq. (6.15), the pth row of the tableau is simply divided by the pivot element apq. Equations (6.16) and (6.17) perform the elimination step in the qth column of the tableau. Elements in the qth column above and below the pth row are reduced to zero by the elimination process thus eliminating xq from all the rows except the pth row. These equations may be coded into a computer program to perform the pivot step. On completion of the pivot step, a new canonical form for the equation **Ax = b** is obtained; i.e., a new basic solution of the equations is obtained. The process of interchanging roles of two variables is illustrated in Example 6.5.

EXAMPLE 6.5

Pivot Step—Interchange of Basic and Nonbasic Variables

Assuming x_3 and x_4 as basic variables, Example 6.3 is written in the canonical form as follows: minimize $f = -4x_1 - 5x_2$ subject to $-x_1 + x_2 + x_3 = 4$, $x_1 + x_2 + x_4 = 6$, $x_i \geq 0$; $i = 1$ to 4. Obtain a new canonical form by interchanging the roles of x_1 and x_4, i.e., make x_1 a basic variable and x_4 a nonbasic variable.

TABLE 6-5. Pivot Step to Interchange Basic Variable X_4 with Nonbasic Variable X_1 for Example 6.5**Initial canonical form.***Basic*↓$X_1X_2X_3X_4$**b**1 x_3-111042 $x_4$11016Basic solution: Nonbasic variables: $x_1 = 0$, $x_2 = 0$Basic variables: $x_3 =4$, $x_4 = 6$To interchange x_1 with x_4, choose row 2 as the pivot row and column 1 as the pivot column. Perform elimination using a_{21} as the pivot element.**Result of the pivot operation: second canonical form**.*Basic*↓$X_1X_2X_3X_4$**b**1 $x_3$0211102 $x_1$11016Basic solution: Nonbasic variables: $x_2 = 0$, $x_4 = 0$Basic variables: $x_1 = 6$, $x_3 = 10$

Solution. The given canonical form can be written in a tableau as shown in Table 6-5; x_1 and x_2 are nonbasic and x_3 and x_4 are basic, i.e., $x_1 = x_2 = 0$, $x_3 = 4$, x_4.= 6. This corresponds to point A in Fig. 6-2. In the tableau, the basic variables are identified in the leftmost column and the rightmost column gives their values. Also, the basic variables can be identified by examining columns of the tableau. The variables associated with the columns of the identity matrix are basic; e.g., variables x_3 and x_4 in Table 6-5. Location of the positive unit element in a basic column identifies the row whose right side parameter b_i is the current value of the basic variable associated with that column. For example, the basic column x_3 has unit element in the first row, and so x_3 is the basic variable associated with the first row. Similarly, x_4 is the basic variable associated with row 2.

To make x_1 basic and x_4 a nonbasic variable, one would like to make $a_{21}' = 1$ and $a_{11}' = 0$. This will replace x_1 with x_4 as the basic variable and a new canonical form will be obtained. The second row is treated as the pivot row, i.e., $a_{21} = 1$ ($p = 2$, $q = 1$) is the pivot element. Performing Gauss-Jordan elimination in the first column with $a_{21} = 1$ as the pivot element, we obtain the second canonical form as shown in Table 6-5. For this canonical form, $x_2 = x_4 = 0$ are the nonbasic variables and $x_1 = 6$ and $x_3 = 10$ are the basic variables. Thus, referring to Fig. 6-2, this pivot step results in a move from the extreme point A(0, 0) to an adjacent extreme point D(6, 0).

> Read full chapter

Linear Programming Methods for Optimum Design

Jasbir S. Arora, in Introduction to Optimum Design (Third Edition), 2012

8.4.2 The Pivot Step

In the Simplex method, we want to systematically search among the basic feasible solutions for the optimum design. *We must have a basic feasible solution to initiate the* Simplex method. Starting from the basic feasible solution, we want to find another one that decreases the cost function. This can be done by interchanging a current basic variable with a nonbasic variable. That is, a current basic variable is made nonbasic (i.e., reduced to 0 from a positive value), and a current nonbasic variable is made basic (i.e., increased from 0 to a positive value). The *pivot step* of the Gauss-Jordan elimination method accomplishes this task and results in a new canonical form (general solution), as explained in Example 8.5. The definitions of pivot column, pivot row, and *pivot element* are also given.

Example 8.5

Pivot Step—Interchange of Basic and Nonbasic Variables

The problem in Example 8.3 is written as follows in the standard form with the linear system **Ax=b** in the canonical form:Minimize(a)subject to(b)(c)(d)Obtain a new canonical form by interchanging the roles of the variables x_1 and x_4 (i.e., make x_1 a basic variable and x_4 a nonbasic variable).

Solution

The given canonical form can be written as the initial tableau, as shown in Table 8.4; x_1 and x_2 are nonbasic, and x_3 and x_4 are basic ($x_1=x_2=0$, $x_3=4$, $x_4=6$). This corresponds to point A in Figure 8.2. In the tableau, the basic variables are identified in the leftmost column, and their values are given in the rightmost column. Also, the basic variables can be identified by examining the columns of the tableau. The variables associated with the columns of the identity submatrix are basic (e.g., variables x_3 and x_4 in Table 8.4). The location of the positive unit element in a basic column identifies the row whose right side parameter *bi* is the current value of the basic variable associated with that row. For example, the basic column x_3 has a unit element in the first row, so x_3 is the basic variable associated with the first row, having a value of 4. Similarly, x_4 is the basic variable associated with row 2 having a value of 6.*Pivot column:* Since x_1 is to become a basic variable, it should become an identity column; that is, x_1 should be eliminated from all rows in the x_1 column except one. Since x_4 is to become a nonbasic variable and is associated with row 2, the unit element in the x_1 column should appear in row 2. This is achieved by eliminating x_1 for row 1 (i.e., Eq. (b)). The column in which eliminations are performed is called the pivot column.*Pivot row:* The row that is used to perform elimination of a variable from various equations is called the pivot row (e.g., row 2 in the initial tableau in Table 8.4).*Pivot element:* The intersection of the pivot column and the pivot row determines the pivot element (e.g., $a_{21}=1$ for the initial tableau in Table 8.4; the pivot element is boxed).*Pivot operation:* To make x_1 a basic variable and x_4 a nonbasic variable, we need to make $a_{21}=1$ and $a_{11}=0$. This will replace x_1 with x_4 as the basic variable and a new canonical form will be obtained. Performing Gauss-Jordan elimination in the first column with $a_{21}=1$ as the pivot element, we obtain the second canonical form as shown in Table 8.4 (add row 2 to row 1). For this canonical form, $x_2=x_4=0$ are the nonbasic variables and $x_1=6$ and $x_3=10$ are the basic variables. Thus, referring to Figure 8.2, this pivot step results in a move from the extreme point A(0, 0) to an adjacent extreme point D(6, 0).

Table 8.4. Pivot step to interchange basic variable x_4 with nonbasic variable x_1 for Example 8.5

Initial tableau

Basic↓	x_1	x_2	x_3	x_4	**b**
1.x_3	–1	1	1	0	4
2.x_4		1	0	1	6

Basic solution: Nonbasic variables: $x_1=0$, $x_2=0$; basic variables: $x_3=4$, $x_4=6$

To interchange x_1 with x_4, choose row 2 as the pivot row and column 1 as the pivot column. Perform elimination using a_{21} as the pivot element.

Second tableau: Result of the pivot operation

Basic↓	x_1	x_2	x_3	x_4	**b**
1.x_3	0	2	1	1	10
2.x_1	1	1	0	1	6

Basic solution: Nonbasic variables: $x2=0$, $x4=0$; basic variables: $x1=6$, $x3=10$

> Read full chapter

More on Linear Programming Methods for Optimum Design

Jasbir S. Arora, in Introduction to Optimum Design (Second Edition), 2004

7.1 7.1 Derivation of the Simplex Method

In the previous chapter, we presented the basic ideas and concepts of the Simplex method. The steps of the method were described and illustrated in several examples. In this section, we describe the theory that leads to the steps used in the example problems.

7.1.1 Selection of a Basic Variable That Should Become Nonbasic

Derivation of the Simplex method is based on answering the two questions posed earlier: (1) which current nonbasic variable should become basic, and (2) which current basic variable should become nonbasic. We will answer the second question in this section. Assume for the moment that xr is a nonbasic variable tapped to become basic. This indicates that the rth non-basic column should replace some current basic column. After this interchange, there should be all zero elements in the rth column except a positive unit element at one location.

To determine a current basic variable that should become nonbasic, we need to determine the pivot row for the elimination process. This way the current basic variable associated with that row will become nonbasic after the elimination step. To determine the pivot row, we transfer all the terms associated with the current nonbasic variable xr (tapped to become basic) to the right side of the canonical form of Eq. (6.13). The system of equations becomes:

(7.1)

Since xr is to become a basic variable, its value should become nonnegative in the new solution. The new solution must also remain feasible. The right sides of Eq. (7.1) represent values of the basic variables for the next Simplex iteration once xr is assigned a value greater than or equal to 0. An examination of these right sides shows that xr cannot increase arbitrarily. The reason is that if xr becomes arbitrarily large, then some of the new right side parameters $(bi - ai_{,r}\, xr)$, $i = 1$ to m may become negative. Since right side parameters are the new values of the basic variables, the new basic solution will not be feasible. Thus for the new solution to be basic and feasible, the following constraints must be satisfied by the right sides of Eq. (7.1) in selecting a current basic variable that should become nonbasic (i.e., attain zero value):

(7.2)

Any *ai,r* that are nonpositive pose no limit on how much *xr* can be increased since Inequality (7.2) remains satisfied; recall that *bi* ≥ 0. For a positive *ai,r*, *xr* can be increased from zero until one of the inequalities in Eq. (7.2) becomes active, i.e., one of the right sides of Eq. (7.1) becomes zero. A further increase would violate the nonnegativity conditions of Eq. (7.2). Thus, the maximum value that the incoming variable *xr* can take is given as

(7.3)

where *s* is the index of the smallest ratio. Equation (7.3) says that we take ratios of the right side parameters *bi* with the positive elements in the *r*th column (*air*'s) and we select the row index *s* giving the smallest ratio. *In the case of a tie, the choice for the index s is arbitrary among the tying indices and in such a case the resulting basic feasible solution may be degenerate.* Thus, Eq. (7.3) identifies a row with the smallest ratio *bi*/*ai,r*. The basic variable *xs* associated with this row should become nonbasic. *If all ai,r are nonpositive in the r*th column, *then xr can be increased indefinitely.* This indicates that the LP problem is *unbounded.* Any practical problem with this situation is not properly constrained so the problem formulation should be reexamined.

7.1.2 Selection of a Nonbasic Variable That Should Become Basic

We now know how to select a basic variable that should replace a nonbasic variable. To answer the first question posed earlier, let us see how we can identify the nonbasic variable that should become basic. *The main idea of bringing a nonbasic variable into the basic set is to improve the design, i.e., to reduce the current value of the cost function.* A clue to the desired improvement is obtained if we examine the cost function expression. To do this we need to write the cost function in terms of the nonbasic variables only. We substitute for the current values of basic variables from Eq. (6.13) into the cost function to eliminate the basic variables. Current values of the basic variables are given in terms of the nonbasic variables as follows:

(7.4)

Substituting Eq. (7.4) into the cost function expression in Eq. (6.7) and simplifying, we obtain an expression for the cost function in terms of the nonbasic variables (*xj*, *j* = *m* + 1 to *n*) as

(7.5)

where f_0 is the current value of the cost function given as

(7.6)

and the parameters *c'j* are

(7.7)

The cost coefficients $c'j$ of the nonbasic variables play a key role in the Simplex method and are called the reduced or relative cost coefficients. They are used to identify a nonbasic variable that should become basic to reduce the current value of the cost function. Expressing the cost function in terms of the current nonbasic variables is a key step in the Simplex method. We have seen that this is not difficult to accomplish because the Gaussian elimination steps can be used routinely on the cost function expression to eliminate basic variables from it. Once this has been done, the reduced cost coefficients $c'j$ can be readily identified.

In general the reduced cost coefficients $c'j$ of the nonbasic variables may be positive, negative, or zero. Let one of $c'j$ be negative. Then, note from Eq. (7.5) that if a positive value is assigned to the associated nonbasic variable (i.e., it is made basic), the value of f will decrease. If more than one negative $c'j$ is present, a widely used rule of thumb is to choose the non-basic variable associated with the smallest $c'j$ (i.e., negative $c'j$ with the largest absolute value) to become basic. Thus, if any $c'j$ for $(m + 1) \leq j \leq n$ (for nonbasic variables) is negative, then it is possible to find a new basic feasible solution (if one exists) that will further reduce the cost function. If a $c'j$ is zero, then the associated nonbasic variable can be made basic without affecting the cost function value. If all $c'j$ are nonnegative, then it is not possible to reduce the cost function any further and the current basic feasible solution is optimum. These results have been summarized previously in Theorems 6.3 and 6.4.

Note that when all $c'j$ in the nonbasic columns are positive, the optimum solution is unique. If at least one $c'j$ (reduced cost coefficient associated with a nonbasic variable) is zero, then there is a possibility of alternate optima. If the nonbasic variable associated with a zero reduced cost coefficient can be made basic according to the foregoing procedure, the extreme point corresponding to alternate optima can be obtained. Since the reduced cost coefficient is zero, the optimum cost function value will not change. Any point on the line segment joining the optimum extreme points also corresponds to an optimum. Note that these optima are global as opposed to local, although there is *no distinct global optimum.* Geometrically, multiple optima for an LP problem imply that the cost function hyperplane is parallel to one of the active constraint hyperplanes.

Note that if the nonbasic variable associated with the negative reduced cost coefficient $c'j$ cannot be made basic (e.g., when all aij in the $c'j$ column are negative), then the *feasible region is unbounded.*

> Read full chapter

More on Linear Programming Methods for Optimum Design

Jasbir S. Arora, in Introduction to Optimum Design (Third Edition), 2012

9.1.6 The Simplex Algorithm

The steps of the Simplex method were illustrated in Example 8.7 with only "≤ type" constraints. They are summarized for the general LP problem as follows:

Step 1. *Problem in the standard form.* Transcribe the problem into the standard LP form.
Step 2. *Initial basic feasible solution.* This is readily available if all constraints are "≤ type" because the slack variables are basic and the real variables are nonbasic. If there are equality and/or "≥ type" constraints, then the two-phase Simplex procedure must be used. Introduction of artificial variable for each equality and "≥ type" constraint gives an initial basic feasible solution to the Phase I problem.
Step 3. Optimality check: The cost function must be in terms of only the nonbasic variables. This is readily available when there are only "≤ type" constraints. For equality and/or "≥ type" constraints, the artificial cost function for the Phase I problem can also be easily transformed to be in terms of the nonbasic variables.
If all of the reduced cost coefficients for nonbasic variables are non-negative (≥0), we have the optimum solution (end of Phase I). Otherwise, there is a possibility of improving the cost function (artificial cost function). We need to select a nonbasic variable that should become basic.
Step 4. *Selection of a nonbasic variable to become basic.* We scan the cost row (the artificial cost row for the Phase I problem) and identify a column having negative reduced cost coefficient because the nonbasic variable associated with this column should become basic to reduce the cost (artificial cost) function from its current value. This is called the pivot column.
Step 5. *Selection of a basic variable to become nonbasic.* If all elements in the pivot column are negative, then we have an unbounded problem. If there are positive elements in the pivot column, then we take ratios of the right-side parameters with the positive elements in the pivot column and identify a row with the smallest positive ratio according to Eq. (9.11). In the case of a tie, any row among the tying ratios can be selected. The basic variable associated with this row should become nonbasic (i.e., zero). The selected row is called the pivot row, and its intersection with the pivot column identifies the *pivot element.*

Step 6. *Pivot step*. Use the Gauss-Jordan elimination procedure and the pivot row identified in Step 5. *Elimination must also be performed in the cost function (artificial cost) row* so that it is only in terms of nonbasic variables in the next tableau. This step eliminates the nonbasic variable identified in Step 4 from all of the rows except the pivot row; that is, it becomes a basic variable.
Step 7. *Optimum solution*. If the optimum solution is obtained, then read the values of the basic variables and the optimum value of the cost function from the tableau. Otherwise, go to Step 3.

> Read full chapter

Sequential Optimization Methods

B. Dejaegher, Y.V. Heyden, in Comprehensive Chemometrics, 2009

1.17.2.2.2(ii) Three- or more-dimensional simplex method

The abovedescribed (basic) simplex method for two factors can be generalized to an f-factor case. Although the two-dimensional simplex can be visualized geometrically, this is no longer possible for three- or more-dimensional simplexes. Nevertheless, the principle remains exactly the same. When examining f factors, the simplex contains $f + 1$ vertices or points. After determination of the vertex to be rejected, the coordinates of the new vertex are obtained as follows. The coordinates of the f retained vertices are summed for each factor and divided by $f/2$. From the resulting values, the coordinates of the rejected point are subtracted, yielding the coordinates of the new vertex. The above are generalizations of Equations (4–5).

In vector notation, when examining f factors, the $f + 1$ points of the initial simplex, B, N_1, N_2, …, N_{f-1}, and W can be represented by the vectors $\mathbf{b}$, $\mathbf{n1}$, $\mathbf{n2}$, …, $\mathbf{n}_{f-1}$ and $\mathbf{w}$, that is, $\mathbf{b} = [x_{1b}, x_{2b}, \ldots, x_{fb}]$, $\mathbf{n1} = [x1_{n_1}, x_{2n_1}, \ldots, x_{fn_1}]$, $\mathbf{n2} = [x1_{n_2}, x2_{n_2}, \ldots, x_{fn_2}]$, …, $\mathbf{n}_{f-1} = [x_{1nf-1}, x_{2nf-1}, \ldots, x_{fnf-1}]$, and $\mathbf{w} = [x_{1w}, x_{2w}, \ldots, x_{fw}]$. B and W are the best and the worst results, respectively. N_1, N_2, …, N_{f-1} are neither the best nor the worst results. The centroid P_1 of the hyperspace remaining after deleting the vertex giving the worst response W, and represented as vector $\mathbf{p1}$, is then defined as given in Equation (6), and the coordinates of the new vertex R_1, represented as vector $\mathbf{r1}$, are given by Equation (5). Furthermore, the same rules, as given above, are valid.

(6)

> Read full chapter

The Minimum Evolution Problem in Phylogenetics: Polytopes, Linear Programming, and Interpretation

Stefan Forcey, ... William Sands, in Algebraic and Combinatorial Computational Biology, 2019

10.3.1 Discrete Integer Linear Programming: The Branch and Bound Algorithm

The problem of finding the vertex of the BME polytope which corresponds to the tree that minimizes our product belongs to a class of problems called discrete integer linear programs. That is the primary reason that we scaled the values in the solution vector to become integer powers of two. One of the most common techniques for solving this class of problems is called *branch and bound*. This process is recursive, breaking the original problem into subproblems, which are easier to solve. The recursive structure of this process can be visualized as traversing a rooted binary tree, where each node represents an individual linear programming problem.

To begin, the discrete valued constraints on the decision variables are relaxed. This allows us to utilize linear programming algorithms where the decision variables admit a continuum of values. The initial linear programming problem that results from this relaxation is called the *root LP*. Computing its solution allows us to determine the feasibility of the original problem. If the solution to the root LP meets our original restrictions for the decision variables, then the branch and bound routine terminates. Otherwise, we select a variable according to a *branching rule* and begin the branching process. The branching rule tells us how to divide the solution space, which results in a set of subproblems, which represent new nodes in our branch and bound tree. After separately solving each of these problems, a *selection strategy* is used to determine which nodes to explore in the branch and bound tree. If a node is not explored, we say the node was *fathomed* or *pruned*. Once a node is selected for exploration, the process is repeated.

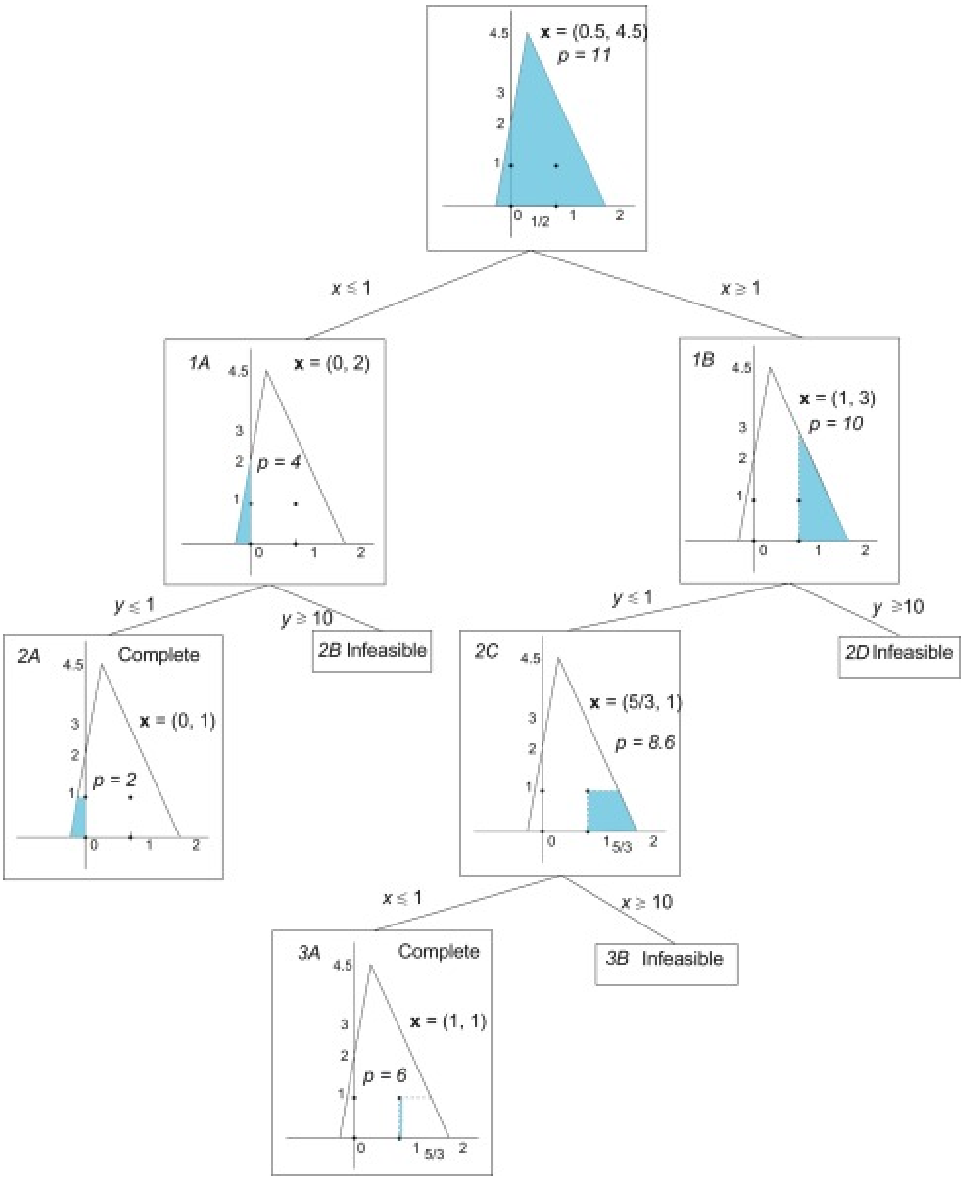
The inequalities used in the creation of individual problems, along the path, are maintained throughout the search. Once a feasible solution satisfying the constraints on the decision variables is obtained, we can update the global bound on the objective and use it to prune subproblems, which provide a less optimal objective value. We are permitted to prune subproblems in this manner, even if the discrete constraints on the decision variables are not satisfied. We call the best current solution, the *incumbent solution*. This pruning process allows us to eliminate significant portions of the search space, which effectively reduces the algorithm's running time. Repeatedly applying this process allows us to eventually obtain the optimal solution to our original discrete programming problem.

To summarize, the steps of the general branch and bound approach are as follows:

1. Run the LP solver (such as the simplex method) on our (relaxed) polytope with our given objective function to get “answer zero”: vector $\mathbf{x}_0$ and objective function value p_0.
2. If $\mathbf{x}_0$ has all coordinates powers of 2, then we say it is *complete*, and it is our final answer.
3. If $\mathbf{x}_0$ is not complete, then we create some new LP problems 1*A*, 1*B*, etc. by adding new inequalities one at a time, just enough to force an offending coordinate away from its disallowed value.
4. We solve each of these (as long as they are still *feasible*, i.e., as long as the intersection of the inequalities is nonempty) to get answers x_1A, x_1B, p_1A, p_1B, etc.
5. For each new answer we check whether it is complete. If complete, check if its objective function value is better than any seen so far: if it is better then save that solution as the current incumbent solution. If not complete then decide whether it *merits further branching* into more new problems; that is, whether it has an objective function value better than the current best value given by a complete answer. If it is not better, then we prune this branch—that is, we do not branch again. Otherwise we return to step (3).
6. The process ends when no more branching is indicated; and the final answer is the optimal one from among the complete answers found.

For an example, let us solve the following problem: Maximize $p = 4x + 2y$ subject to

Require all coordinates of answer $\mathbf{x} = (x, y)$ to be in the set $\{0, 1, 10\}$.



The uppermost box shows the 0-level problem with its solution. Then the branching is performed in alphabetic order on the variables *x* and *y*. Of course this is an artificial example, in which the solution is easily found and the inequalities are not very helpful! Larger examples are needed to see that the method is efficient.

In this example, some of the branching paths end in an infeasible problem and some end in a complete solution. This example does not have a path that ends in a pruned solution. Next, we give as an exercise a similar problem that does have an opportunity to prune. It also has the same requirement on discrete coordinate values as our actual BME problem.

Exercise 10.6

Perform branch and bound. Maximize $p = 6.75x + 5y$ subject to

Require all coordinates of the answer (x, y) to be powers of 2. When branching, take the first alphabetical coordinate value that is not a power of 2 and introduce two branches that add the inequalities ≤ and ≥ the nearest powers of two smaller and larger than that coordinate value, respectively. This of course is an arbitrary choice of strategy for branching. In the next section we will talk about more tailored strategies for our specific BME problem.

> Read full chapter

Further Topics in Linear Programming

Bernard Kolman, Robert E. Beck, in Elementary Linear Programming with Applications (Second Edition), 1995

3.4 ExERCISES

In Exercises 1–5 the given tableau represents a solution to a linear programming problem that satisfies the optimality criterion, but is infeasible. Use the dual simplex method to restore feasibility.

1.

2.

3.

4.

5.

6. Use the dual simplex method to verify that Tableau 3.27 is correct,
7. For Example 3, verify using the dual simplex method that the final tableau (Tableau 3.30) is correct. Note that your answer may differ slightly from the text due to round-off error.
8. Use the dual simplex method to find a solution to the linear programming problem formed by adding the constraint$3x_1 + 5x_3 \geq 15$to the problem in Example 2
9. Example 3 showed that adding a constraint may change the solution to a linear programming problem (i.e., the new solution has different basic variables and the basic variables have different values). There are two other possibilities that may occur when a constraint is added. Describe them.
10. **Computing project.** Compare the structure diagrams for the simplex algorithm and the dual simplex algorithm. How is duality exemplified by these diagrams?

> Read full chapter

Handbook of Constraint Programming

John N. Hooker, in Foundations of Artificial Intelligence, 2006

15.3.2 Simplex Method

Given a basic feasible solution ($B^{-1}b$, 0), the simplex method can find a basic optimal solution of (15.3) or show that (15.3) is unbounded. If $r \geq 0$, the solution ($B^{-1}b$, 0) is already optimal. Otherwise increase any nonbasic variable x_j with negative reduced cost r_j. If the column of $B^{-1}N$ in (15.4) that corresponds to x_j is nonnegative, then x_j can increase indefinitely without driving any component of x_B negative, which means (15.3) is unbounded. Otherwise increase x_j until some basic variable x_i hits zero. This creates a new basic solution. The column of B corresponding to x_i is moved out of B and the column of N corresponding to x_j is moved in. B^{-1} is quickly recalculated and the process repeated.

The procedure terminates with an optimal or unbounded solution if one takes care not to cycle through solutions in which one or more basic variables vanish (*degeneracy*). A starting basic feasible solution can be obtained by solving a "Phase I" problem in which the objective is to minimize the sum of constraint violations. The starting basic variables in the Phase I problem are temporary slack or surplus variables added to represent the constraint violations that result when the other variables are set to zero.

More than half a century after its invention by George Dantzig, the simplex method is still the most widely used method in state-of-the-art solvers. *Interior point* methods are competitive for large problems and are also available in commercial solvers.

> Read full chapter

Integer Programming

Bernard Kolman, Robert E. Beck, in Elementary Linear Programming with Applications (Second Edition), 1995

Gomory's Method for Mixed Integer Programming

We can attempt to solve a mixed integer programming problem in the same way as a pure integer programming problem. We use the simplex method to obtain the optimal solution to the related linear programming problem. This will be a solution to the mixed integer programming problem if those variables appearing in the basis for the optimal solution that are required to have integer values actually do have such values. We suppose now that x_{ri}, the ith basic variable in the optimal solution of the related linear programming problem is required to be integral and that x_{Bi} is not an integer. We saw that we may write the ith constraint (9) as

(14)

Let $x_{Bi} = [x_{Bi}] + f_i$. We may write (14) as

or

(15)

We now divide the set of indices for the nonbasic variables into two subsets N_+ and N_-, where

$N_+ = \{j \mid j$ is the index of a nonbasic variable and $t_{ij} \leq 0\}$

$N_- = \{j \mid j$ is the index of a nonbasic variable and $t_{ij} < 0\}$.

Then (15) may be written as

(16)

Using ((16)), we want to derive a cutting plane constraint that will cut off the current optimal solution to the related linear programming problem because in that solution x_{ri} does not have an integer value. The cutting plane constraint must also have the property, as before, that no feasible solutions for the mixed integer problem should be cut off. We can interpret this condition as saying that if x_{ri} satisfies (16) and if x_{ri} is an integer, then x_{ri} must satisfy the cutting plane constraint.

We consider two cases: the right-hand side of (16) is positive, or it is negative. Assume, first, that $f_i + ([x_{Bi}] - x_{ri}) < 0$. The quantity in parentheses is an integer by assumption and $0 < f_i < 1$. We see that $[x_{Bi}] - x_{ri}$ must be a negative integer for all these assumptions to hold.

Thus, the largest value that the right-hand side of (16) can have while still remaining negative is $f_i - 1$. Our original constraint (14) implies, in this case,

(17)

We can make the left-hand side of (17) smaller by removing the first sum, since it represents a nonnegative quantity. We obtain

Dividing by $f_i - 1$ and reversing the inequality, since $f_i - 1$ is negative, and then multiplying by f_i yields

(18)

Since the first sum in (17) is nonnegative, we may add it to (18) to obtain

(19)

For the other case, we assume that $f_i + ([x_{Bi}] - x_{ri}) \geq 0$. Using reasoning similar to that in the first case, we see that $[x_{Bi}] - x_{ri}$ must be nonnegative. Thus, our original constraint (14) implies that

(20)

We may replace the second sum in (20) by any larger quantity and still maintain the inequality. We have

Consequently, (20) implies that

which is the same as (19).

We have now shown that if (14) is a constraint in a mixed integer problem whose corresponding basic variable is supposed to have an integer value but does not, then (19) is satisfied by every vector **x** that satisfies (14), assuming that x_{ri} is an integer. From (19) we can construct the equation of the cutting plane. We reverse the inequality in (19) and add a slack variable u_i, obtaining

(21)

Equation (21) is the Gomory cutting plane. For the current optimal solution, $x_j = 0$ if j □ N_+ or j □ N_-. Therefore, introducing the constraint (21) gives

$u_i = -f_i < 0$

for the current optimal solution, and hence it has been cut off, since u_i must be nonnegative.

Note that (21) may be written as

where

(22)

It is possible that some of the nonbasic variables are also required to be integer valued. We can refine the cutting plane defined by (21) if we use this information. The refinement will simply cut off more solutions that do not satisfy the integrality conditions. We write

$t_{ij} = [t_{ij}] + g_{ij},$

Then, instead of the definitions of d_j given in (22), we have for the refined cutting plane

(23)

> Read full chapter