A modification of the simplex method reducing roundoff errors

Ralf Ulrich Kern

Georg-Simon-Ohm-Hochschule
Nürnberg
Fakultät Informatik

Postfach 21 03 20
90121 Nürnberg

Ralf-Ulrich.Kern@ohm-hochschule.de

Abstract

The aim of this paper is to present a modification of the simplex algorithm giving under certain, numerically harder, circumstances surprisingly more precise results than the standard method does. Furthermore the modified method can be combined with Gomory's algorithm in order to solve linear integer optimization problems more easily.

ISSN 1867-5433 Sonderdruck Schriftenreihe der Georg-Simon-Ohm-Hochschule Nürnberg Nr. 45, Nürnberg 2009

1. Introduction

The aim of this paper is to present a modification of the simplex algorithm giving under certain, numerically harder, circumstances surprisingly more precise results than the standard method does.

Consider the linear optimization problem in standard form:

$$\text{Maximize} \quad c^T x$$

$$\textbf{(LP)} \qquad \text{subject to } Ax \le b,\ x \ge 0, x \in \mathbf{R}^n,$$

where $c \in \mathbf{Z}^n$, $A \in \mathbf{Z}^{n\text{x}m}$, $b \in \mathbf{Z}^m$ and $b \ge 0$.

that is, all input data have to be integers. *The solution of this problem, however, is not required to be integer-valued*, that means, we are dealt with a continuous and not an integer problem.

The modified simplex algorithm to be presented has the following effect: Starting with integer input data, every subsequent simplex tableau will have integer values. Only for evaluating the final tableau it may be necessary to perform divisions where non-integer results may occur. This implies that, given a sufficiently large, but (in contrast to using floating point numbers) always finite number of decimal places for the computation, the modified method will not cause any rounding errors - except, perhaps, in the evaluation of the final tableau.

In practical computations, of course, the number of decimal places available for the computation usually is not only finite, but also bounded, such that nevertheless rounding may occur. Anyway, the modified method has the property to prevent the involved numbers from increasing too much in length. This feature will be discussed later on in more detail.

The restriction to integer input data is not very relevant, since problems with fractions can be converted to fit into the integer requirements. Although the modified method deals with continuous optimization problems, it is possible to apply it to integer optimization. This point will also be considered in some detail.

2. Definition of the modified algorithm

The general simplex tableau of the modified algorithm in short (abbreviated) form is given by

$$\begin{array}{c|ccccc|c}
d & x_{k_1} & \cdots & x_{k_j} & \cdots & x_{k_n} & \\
\hline
x_{h_1} & a'_{11} & \cdots & a'_{1j} & \cdots & a'_{1n} & b'_1 \\
\vdots & \vdots & & \vdots & & \vdots & \vdots \\
x_{h_i} & a'_{i1} & \cdots & a'_{ij} & \cdots & a'_{in} & b'_i \\
\vdots & \vdots & & \vdots & & \vdots & \vdots \\
x_{h_m} & a'_{m1} & \cdots & a'_{mj} & \cdots & a'_{mn} & b'_m \\
\hline
 & z'_1 & \cdots & z'_j & \cdots & z'_n & f
\end{array}$$

It differs from the short form tableau of the standard simplex method only by showing one additional term, *the corner divisor d*. Note that the x_{h_i} (i = 1, ..., m) are the basic variables.

The initial tableau of the modified method is defined in the following way:

$$\begin{aligned}
&d \quad := 1, \quad f := 0, \\
&a'_{ij} := a_{ij}, \quad b'_i := b_i, \quad z'_j := -c_j; \\
&x_{k_j} := x_j, \quad x_{h_i} := x_{n+i}, \\
&\qquad\qquad (i = 1, ..., m; j = 1, ..., n),
\end{aligned}$$

where the entries a_{ij}, b_i, c_j are given by the initial problem (LP), yielding the following initial tableau

$$\begin{array}{c|ccccc|c}
1 & x_1 & \cdots & x_j & \cdots & x_n & \\
\hline
x_{n+1} & a_{11} & \cdots & a_{1j} & \cdots & a_{1n} & b_1 \\
\vdots & \vdots & & \vdots & & \vdots & \vdots \\
x_{n+i} & a_{i1} & \cdots & a_{ij} & \cdots & a_{in} & b_i \\
\vdots & \vdots & & \vdots & & \vdots & \vdots \\
x_{n+m} & a_{m1} & \cdots & a_{mj} & \cdots & a_{mn} & b_m \\
\hline
 & -c_1 & \cdots & -c_j & \cdots & -c_n & 0
\end{array}$$

In a modified tableau, the pivot term is chosen among the entries a'_{ij} like in a standard tableau, according to the usual or any other pivot choice rule. After having chosen the pivot term p, the situation can be represented like this:

$$\begin{array}{cccc}
\underline{d|} & x_j & & \\
x_i & \boxed{p} & \cdots & r \\
 & \vdots & & \vdots \\
 & c & \cdots & e
\end{array}$$

where d denotes the corner divisor, p the chosen pivot term, r (c) any other element in the pivot row (column, resp.), x_i (x_j) the corresponding (non-)basic variables (to be exchanged in the pivot step), and e any entry elsewhere in the tableau. The exchange step is now performed according to the **rectangle rule** for the modified tableau:

$$\begin{array}{cccc}
\underline{d|} & x_j & & \\
x_i & \boxed{p} & \cdots & r \\
 & \vdots & & \vdots \\
 & c & \cdots & e
\end{array}
\longrightarrow
\begin{array}{cccc}
\underline{p|} & x_i & & \\
x_j & d & \cdots & r \\
 & \vdots & & \vdots \\
 & -c & \cdots & \frac{pe-rc}{d}
\end{array}$$

After having reached the final tableau, that is when it is no more possible to find a pivot column, all tableau entries (or at least the b_i's at the right hand side) are to be divided by the corner divisor; then the evaluation is done in the same way as with a final tableau of the standard simplex algorithm.

The significance of the modified method is explained by the

Theorem

a) Dividing the whole modified tableau by the corner divisor d yields the same standard simplex tableau that is obtained by executing the standard simplex algorithm starting with the same initial tableau and making the same pivot choices in each exchange step as have been taken with the modified method.

b) The row $\quad x_{h_i} \;|\; a'_{i1} \;\cdots\; a'_{ij} \;\cdots\; a'_{in} \;|\; b'_i$

of a modified tableau represents the equation

$$d' x_{h_i} + a'_{i1} x_{k_1} + \cdots + a'_{ij} x_{k_j} + \cdots + a'_{in} x_{k_n} = b'_i.$$

c) The quotient *(pe - rc)/d* obtained by executing an exchange step in a modified tableau is *always integer-valued*.

Proof:

a) By induction on the sequence of exchange steps. The assertion is obvious for the initial tableau, since d = 1.

For the standard simplex algorithm the usual rectangle rule is used, that is

$$\begin{array}{ccc} \boxed{p'} & \cdots & r' \\ \vdots & & \vdots \\ c' & \cdots & e' \end{array} \longrightarrow \begin{array}{ccc} \frac{1}{p'} & \cdots & \frac{r'}{p'} \\ \vdots & & \vdots \\ -\frac{c'}{p'} & \cdots & e' - \frac{r'c'}{p'} \end{array}$$

Let the situation in an intermediate tableau of the modified method be represented by the following data necessary for the rectangle rule (with p being the pivot term)

$$\begin{array}{l} \underline{d}| \\ \quad \begin{array}{ccc} \boxed{p} & \cdots & r \\ \vdots & & \vdots \\ c & \cdots & e \end{array} \end{array}$$

then the following operations can be commuted:

$$\begin{array}{ccccc}
\begin{array}{l} \underline{d}| \\ \quad \begin{array}{ccc} \boxed{p} & \cdots & r \\ \vdots & & \vdots \\ c & \cdots & e \end{array} \end{array} & \xrightarrow[\text{step}]{\text{exchange}} & \begin{array}{l} \underline{p}| \\ \quad \begin{array}{ccc} d & \cdots & r \\ \vdots & & \vdots \\ -c & \cdots & \frac{pe-rc}{d} \end{array} \end{array} & & \text{(modified tableau)} \\
\Big\downarrow \text{division by } d & & \Big\downarrow \text{division by } p & & \\
\begin{array}{ccc} \boxed{\frac{p}{d}} & \cdots & \frac{r}{d} \\ \vdots & & \vdots \\ \frac{c}{d} & \cdots & \frac{e}{d} \end{array} & \xrightarrow[\text{step}]{\text{exchange}} & \begin{array}{ccc} \frac{d}{p} & \cdots & \frac{r}{p} \\ \vdots & & \vdots \\ -\frac{c}{p} & \cdots & \frac{e}{d} - \frac{rc}{pd} \end{array} & & \text{(standard tableau)}
\end{array}$$

where the arrow “division by d” yields the equivalent standard tableau by induction hypothesis and the arrow “division by p” proves the induction assertion.

b) Using the result just proved, the given row may be divided by the corner divisor d' to yield the corresponding row in an equivalent standard tableau which can be considered to be the equation

$$x_{h_i} + \sum_{j=1}^{n} \frac{a'_{ij}}{d'} x_{k_j} = \frac{b'_i}{d'},$$

proving the assertion.

c) Consider for a given linear optimization problem a sequence of modified tableaus and their corresponding standard tableaus which are obtained by making the same pivot choices in each exchange step. Keep in mind that (according to part *a* of this theorem) the entries of the standard tableau are equal to corresponding entries of the modified tableau divided by the corner divisor d.

Next we show by induction on the sequence of exchange steps that in a modified tableau d is equal to the determinant of the basis of the corresponding standard tableau. This is obviously true for the initial tableau with d = 1 being the determinant of the unit matrix.

Now let some intermediate modified tableau with pivot term p and corner divisor d be given. The corresponding pivot term in the standard tableau will be $p' := p/d$. Let B be the basis of the standard tableau before the exchange step and $\underline{B}$ after it. It is well-known (see e.g. [Gas], chapter 4, section 2) that det $\underline{B}$ = (det B)·p', and by induction hypothesis we have det B = d such that

$$\det \underline{B} = d \cdot p' = d \cdot \frac{p}{d} = p,$$

but after having executed the exchange step, the pivot term p will be the new corner divisor. Now assume that B is the basis of a given standard tableau and that N is the matrix of the remaining nonbasic vectors of this tableau. N and B are part of the initial (long-form) tableau, and therefore both consist of integers only. Then the matrix formed by the coefficients a_{ij} of the abbreviated standard tableau without the right-hand side and the objective function's row is equal to $B^{-1}N$. According to Cramer's rule, each entry of B^{-1} has the form c/(det B) = c/d, where c is some minor of B, thus an integer, and the same form applies to the entries of $B^{-1}N$ as well as on the right-hand side and the objective function's row. Therefore the entries of the corresponding abbreviated modified tableau are integers, since, according to part *a* of the theorem, they are obtained by multiplying the entries of $B^{-1}N$ with d. This eventually proves the assertion. ■

3. Numerical considerations

In practical computations with the modified method the number of decimal places of the tableau entries tend to increase by a certain amount. With respect to this problem, there are three possibilities to do integer arithmetics on a computer:

1. Use an integer data type,
2. use an integer data type with saturation arithmetics,
3. embed integers into a floating point number data type.

The first possibility includes the danger of overflow when exceeding the range of the chosen data type, leading in such a case to premature program termination when the result of an operation exceeds this range. The second one replaces the result in this case by the largest (or smallest, if it is negative) representable integer, so that the program can be continued, but possibly with a significant error.

The third choice behaves similarly as the second one, but with a considerably smaller roundoff error. The floating point standard IEEE 754 (see [Kah] or [IEEE], section 4) implemented in all current microprocessors with floating point unit (e. g. those with x86 architecture) requires that arithmetical operations on integers represented by floating point numbers be precise as long as the results are integer-valued and do not exceed the range of the embeddable integers (i. e. up to 17 decimal places for 64-bit floating point numbers). Therefore a floating point data type should be used for the modified simplex method.

Since in practical computations, as has already been mentioned, the available number of binary (and likewise decimal) places is bounded, rounding-off or overflow (depending on the variable type used) may occur if intermediate results exceed the maximum number length given by the data type range. However, the division by the corner divisor d in the expression $(pe - rc)/d$ slows down the increase of the number of digits of the tableau entries. Nevertheless it is advisable to use another than the usual pivot choice rule (the steepest descent rule), namely the rule resulting in the largest increase of the objective function, because this rule saves exchange steps, thus limiting the increase of the entries' size even more.

When evaluating the expression $(pe - rc)/d$ during an exchange step, it is recommended to compute $pe - rc$ with double precision, because the products pe and rc can have double length, and any rounding occurring in the calculation of $pe - rc$ may destroy the integer property of $(pe - rc)/d$. Dividing $pe - rc$ by d usually restores normal length such that single length storage might be sufficient.

The well-known method of scaling applied to the modified tableau can destroy the integer property and therefore is not admissible. The initial tableau, however, should not contain common divisors greater than 1 in any row or column in order to keep the entries small. But even if all common divisors have been cancelled in the initial tableau, in the course of the following exchange steps it may happen that there are common divisors of the whole tableau (including the corner divisor). Cancelling such common divisors would reduce the number of decimal places needed for the tableau entries, but also can destroy the integer property of the next tableau. However, the following rule can be stated:

Cancellation rule

If the whole pivot row or the whole pivot column is divisible by the *corner* divisor *d*, then a *common* divisor *t* of the whole tableau (including the corner divisor) can be cancelled without risking the all-integer-property.

Proof:

Denote the *common* divisor of the whole tableau by *t*, and assume that, say, the pivot row is divisible by *d*. (The proof for the pivot column case is almost verbatim the same.) Then $\overline{p}$:= *p/d* and $\overline{r}$:= *r/d* are integers, and executing the exchange step in the cancelled tableau yields according to our rectangle rule:

$$e' = \frac{\frac{p}{t}\frac{e}{t} - \frac{r}{t}\frac{c}{t}}{\frac{d}{t}} = \frac{1}{t}\frac{pe - rc}{d} = \bar{p}\frac{e}{t} - \bar{r}\frac{c}{t} \in \mathbb{Z}.$$

■

If the conditions of the cancellation rule are met, it is advisable to cancel a common divisor of the whole tableau before executing the next exchange step in order to reduce the length of the tableau entries.

4. Application to linear integer optimization problems

It is well-known that GOMORY's first algorithm [Gom] for the solution of linear *integer* optimization problems is very sensitive to rounding errors. GOMORY did recognize this problem and therefore devised a second method to overcome it. Now we have the possibility to combine the modified simplex method with GOMORY's first algorithm in order to let it operate like GOMORY's second one, but in an easier way. In the sequel the combined method will be outlined, following GOMORY's idea of constructing additional cutting planes.

Let the following linear integer optimization problem be given:

Maximize $c^T x$

(IP) subject to $Ax \le b$, $x \ge 0$, $x \in \mathbf{Z}^n$,

where $c \in \mathbf{Z}^n$, $A \in \mathbf{Z}^{n\mathrm{x}m}$, $b \in \mathbf{Z}^m$ and $b \ge 0$.

Assume that the relaxed problem has been solved, neglecting the integer constraint, by means of the modified simplex method. Assume further that in the final tableau there is an index i_0 with $1 \le i_0 \le m$ such that *d'* is not a divisor of b'_{i_0}, where *d'* is the corner divisor and b'_{i_0} the i_0-th entry of the right-hand side of the final tableau (if there is no such i_0, the solution is already the optimal integer solution). Then the i_0-th row of the final tableau in the form

$$x_{h_{i_0}} \quad \Big| \quad a'_{i_0 1} \quad \cdots \quad a'_{i_0 j} \quad \cdots \quad a'_{i_0 n} \quad \Big| \quad b'_{i_0}$$

represents according to part *b* of the theorem the following equation satisfied by all feasible points $x \in \mathbf{R}^n$:

$$(*) \qquad d' x_{h_{i_0}} + \sum_{j=1}^{n} a'_{i_0 j} x_{k_j} = b'_{i_0}.$$

With the definitions

$$\begin{aligned} \bar{a}_{i_0 j} &:\equiv a'_{i_0 j} \pmod{d'} \quad \text{with} \quad 0 \le \bar{a}_{i_0 j} < d', \\ \bar{b}_{i_0} &:\equiv b'_{i_0} \pmod{d'} \quad \text{with} \quad 0 < \bar{b}_{i_0} < d', \\ A_{i_0 j} &:= \frac{a'_{i_0 j} - \bar{a}_{i_0 j}}{d'} \in \mathbb{Z}, \\ B_{i_0} &:= \frac{b'_{i_0} - \bar{b}_{i_0}}{d'} \in \mathbb{Z}, \end{aligned}$$

the equation (*) can be represented as follows:

$$d' x_{h_{i_0}} + \sum_{j=1}^{n} (\bar{a}_{i_0 j} + d' A_{i_0 j}) x_{k_j} = d' B_{i_0} + \bar{b}_{i_0}$$

and hence

$$d' x_{h_{i_0}} + d' \sum_{j=1}^{n} A_{i_0 j} x_{k_j} - d' B_{i_0} = \underbrace{-\sum_{j=1}^{n} \bar{a}_{i_0 j} x_{k_j}}_{\le 0} + \bar{b}_{i_0} \le \bar{b}_{i_0} < d'.$$

Since, for all $x \in \mathbf{Z}^n$, the left-hand side of this equation is divisible by d', this holds true for the right-hand side as well, such that

$$1 > \frac{-\sum_{j=1}^{n} \bar{a}_{i_0 j} x_{k_j} + \bar{b}_{i_0}}{d'} \in \mathbb{Z}.$$

Therefore the new GOMORY constraint is satisfied for all feasible points $x \in \mathbf{Z}^n$:

$$-\sum_{j=1}^{n} \bar{a}_{i_0 j} x_{k_j} + \bar{b}_{i_0} \le 0.$$

On the other hand, the new GOMORY constraint cannot be satisfied by the relaxed optimal solution of the last tableau, because all x_{k_j} are non-basic variables there and therefore 0. Adding this new constraint to the last tableau then reduces the feasibility region without cutting off integer-valued feasible points. Before continuing the computation it is advisable to dualize the tableau yielding that the next pivot term (and hence the next corner divisor) is to be found among the $a_{i_0 j}$ which are < d'. This implies that the sequence of corner divisors is strictly decreasing when adding further GOMORY's constraints. The procedure ends after a finite number of steps, as the corner divisors have to be positive integers and, if the method has not finished before, the last corner divisor will be 1 yielding an optimal integer solution. Furthermore, the corner divisor of the final tableau of the relaxed problem gives an upper bound for the number of additional GOMORY constraints. ■

This consideration also indicates a sound choice rule if in one step more than one GOMORY constraint are eligible (that is if there are several b'_{i_0} which are not divisible by d'): Take among the possible GOMORY constraints the one yielding the smallest pivot term. This effects (at least on a short range optimization horizon) that the sequence of corner divisors is decreasing most rapidly.

Since the coefficients of a GOMORY constraint differ only by multiples of the actual corner divisor from the corresponding entries of the row from which the considered GOMORY constraint has been derived, it follows that part *c* of the theorem still remains true (and hence the all-integer property is maintained) after adding GOMORY constraints.

5. Conclusion

A modification of the simplex method has been presented that costs, apart from some computations with double precision, not more than the standard method does, but can achieve a considerable gain in accuracy.

6. References

[Dan] DANTZIG, G. B.: Linear programming and extensions. Princeton 1963.

[Gas] GASS, S. I.: Linear programming, methods and applications. New York 1969.

[Gom] GOMORY, R. E.: An algorithm for integer solutions to linear programs, in: Graves, P. H.; Wolfe, P. (ed.): Recent advances in mathematical programming. New York, San Francisco, Toronto, London 1963.

[Kah] KAHAN, W.: Lecture Notes on the Status of IEEE Standard 754 for Binary Floating-Point Arithmetic. University of California Berkeley CA. Oct. 1, 1997 http://www.cs.berkeley.edu/~wkahan/ieee754status/IEEE754.PDF

[IEEE] IEEE Standard for Binary Floating-Point Arithmetic IEEE 1985 http://754r.ucbtest.org/standards/754.pdf

[MMe] MÜLLER-MERBACH, H.: On round-off errors in linear programming. Lecture Notes in Operations Research and Mathematical Systems no. 37. Berlin - Heidelberg - New York 1970.

[Nm-M] NEUMANN, K.; MORLOCK, M.: Operations Research. München 2002[2].

[Zur] ZURMÜHL, R.: Matrizen. Berlin - Heidelberg - New York 1964.

[ZuF] ZURMÜHL, R.; FALK, S.: Matrizen 1 - Grundlagen. Berlin - Heidelberg - New York 1997[7].