CHAINED VECTOR SIMPLEX

S. Khanmohammadi

Electrical Engineering Department
University of Tabriz
Tabriz, Iran

Received May 1987

Abstract An algorithm for solving linear programming problems whose matrix of coefficients contains a large number of "zero" entries is studied. This algorithm is more useful when it is generated as a sub-program in a real-time program. The singly linked lists for storing only the non-zero entries of the coefficient matrix is used. The modified Revised Simplex Method is also used for solving such problems because of its advantages.

چکیده در این مقاله ، یک الگوریتم برای حل مسایل برنامه‌ریزی خطی که در آن ماتریس ضرایب تعداد زیادی عنصر "صفر" دارد ، مورد مطالعه قرار گرفته است. این الگوریتم ، وقتی که در یک مسئله زمان حقیقی به صورت یک برنامه فرعی نوشته شود مفیدتر است . برای ذخیره فقط عناصر صفر ماتریس ضرایب ، از فهرستهای تک پیوندی استفاده شده است . برای حل این نوع مسایل نیز از روش سیمپلکس تجدید نظر شده ، به خاطر محاسنی که دارد ، استفاده گردیده است .

INTRODUCTION

During the last decade, linear programming problems and the development of the Simplex Method for its solution has gained many engineering applications. Thus many efforts are going on into the possible techniques through which linear programming can be carried out in a rapid, accurate and efficient manner. For this reason, the linear linked lists (chains) have been devloped largely in recent years, for economizing storage locations in large scale problems.

This paper introduces an algorithm, by utilizing properties of linked lists in linear programming problems, in which the coefficient matrix contains a large number of "zero" entries. A general notion on the singly linear linked lists (single chains), and also a multiplication formula, for multiplying a vector and a chain is given in the first section. The second section emphasizes on the principles of the Revised Simplex Method (R. S. M) and its advantages. Finally in the last section the modified Revised Simplex algorithm for solving the linear programming problems has been developed, where the data is listed in the singly linear linked lists.

LINEAR LINKED LISTS

An array is an ordered aggregation of homogeneous data units, arranged in the memory locations so that physical order is the same as logical order (i.e. there is the first and the last entries). Each entry of the array (except the last) has a successor, which is the logical order and is denoted by the subscripted value. The place of any entry in the memory locations is a simple funciton of the place of the entry and its subscript. A typical example is shown in Figure 1. A linked list is also an ordered aggregation of homogeneous data units, but the physical and logical orders are not the same. Therefore the logical order must be recorded in the memory locations together with the other informations. The order is described by links attached to each array entry which is jointly called a node (Figure 2).

array entry	value	place in the memory locations
y(1)	0.	100
y(2)	5.	101
y(3)	0.	102
y(4)	0.	103
y(5)	10.	104
y(6)	0.	105
"	"	"
"	"	"
"	"	"
y(11)	24.	110
y(12)	0.	111
"	"	"
"	"	"
"	"	"
y(100)	0.	199

Figure 1

Information	Link

Figure 2

A singly linked list (single chain) is a series of nodes linked together. The first node is often called the head of the chain, while the last is called the tail. The linked lists have two advantages: a) It is easy to insert or delete a new entry in the middle of a list (chain), b) It is possible to store only the necessary entries of the list (e.g. non-zero entries) which is used in this paper.

CHAINED VECTOR

We call a chained vector, a singly linked (single chained) representation of a vector. Each node i of a chined vector contains the following informations:

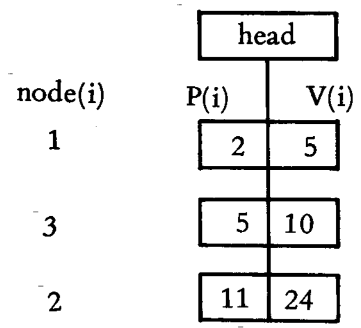
P(i), The index (position) of the node i in the principal vector;

V(i), The value of the node i;

L(i), The link of the node i.

The chained vectors are shown in italics. Also the arrow-and-box diagram representation for the chained vectors have been utilized. In the programs, the chained vectors are represented by three vectors, each of which holding one of the above information for the nodes 1 and 2.

In the following figure, the arrow-and-box diagram (Figure 3-a), and vector representation (Figure 3-b) are given to represent the non-zero entries of vector y of Figure 1 in the chained vector form Y.



3-b

node	head	1	2	3
P(i)	—	2	11	5
V(i)	—	5	24	10
l(i)	1	3	0	2

3-a

Figure 3

Vector and Chain Multiplication

The product of two vectors A and B can be obtained by the following fromula:

$$A \cdot B = \sum_i a_i \cdot b_i \qquad (1)$$

where a_i and b_i are the i^{th} entries of vectors A and B respectively.

By noting the usual rule for dot product, the dot product of a vector A and a chained vector B can be obtained as follows:

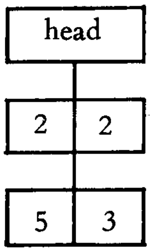
$$A \cdot B = \sum_i a_{p(i)} \cdot v(i) \qquad (2)$$

where $a_{p(i)}$ is the $p(i)^{th}$ entry of the vector A, and V(i) is the value of the node i of the chained vector B.

Example 1:

$$[1,2,4,6,8,10] \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \\ 3 \\ 0 \end{bmatrix} = 1\,x_0+2x_2+4x_0+6x_0+8x_3+10x_0=28$$

[1,2,4,6,8,20].



$=2x_2+8x_3=28$

Simplex and Revised Simplex Methods

The Simplex algorithm is a basic method for solving linear programming problems [3], but its utilization needs a large number of memory locations of the computer. Thus the R. S. M [4], [5] and especially its variation which employs the product form of the inverse is used in high speed computers [6], [7], [8].

There are two major advantages for R. S. M: i) It always deals with the original coefficients (the entries of each column of the coefficient matrix at a time), ii) We need only to record the inverse matrix and the solution vector (m+1) (m+2) arrays instead of an (n+m+1) (m+2) array for the Simplex Method, where n is the number of variables and m is the number of constraints. Mokhtar S. Bazdraa and John J. Arvis have given the number of multiplications (division is considered as a multiplication) and additions (subtraction is considered as an addition) per iteration of both procedures in Table (1) [8].

Chained Vector Simplex Algorithm

In the R. S. M. the original data units are

Table 1. Comparison of Simplex and Revised Simplex Methods.

Method		Operation		
		Pivoting	$z_j - c_j'$ s	Total
Simplex	Mult.	(m+1) (n−m+1)		m(n−m)+n+1
	Add.	m(n−m+1)		m(n−m+1)
R. S. M	Mult.	$(m+1)^2$	m(n−m)	$m(n-m)+(m+1)^2$
	Add.	m(m+1)	m(n−m)	m(n+1)

recorded on the disk and transferred into the memory whenever they are needed. This data transferring is normally very slow and leads to many disadvantages, especially when the R. S. M is used as a subprogram in a real-time program for large scale linear programming problems. This is because at each call we need to generate, and record these data on the disk and transfer then into the memory.

The modified Chained Vector Simplex (C. V. S.) algorithm, have been developed in order to avoid these inconveniences. In this algorithm, only the non-zero entries of the coefficient matrix A is generated and recorded at each call. It is clear that the efficiency of the algorithm depends on the number of non-zero entries of the matrix A. The non-zero entries are generated and recorded as a set of singly linked lists (chained vectors), where each column j of the matrix A is represented with a chained vector A_j. In this way instead of recording all n x m elements of matrix A, 3p+n entries are recorded, where p is the number of non-zero entries of matrix A.

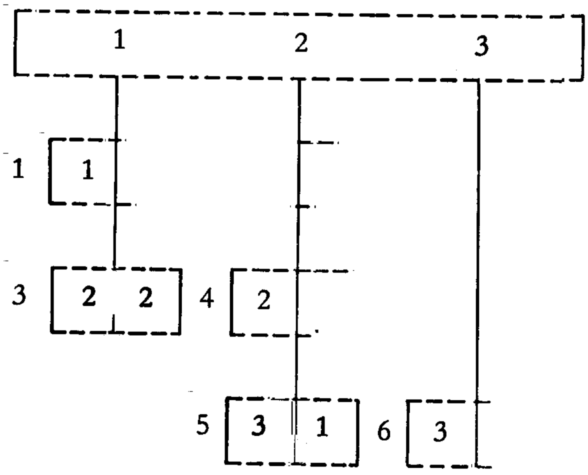
There are two algorithms with two phases. In the first phase of the first algorithm, the non-zero entries of matrix A and the entries of the Right Hand Side (R. H. S) vectors are generated and stored. In the second phase, the standard form of the problem is set up by generating and inserting the slack variables (for the constraints of the form = <), the surplus variables (for the constraints of the form > =), the artificial variables and the cost coefficients. Figures 4 and 5 show the arrow and box and vector representations of the matrix A and the R. H. S vector for the following example and its standard form.

Example 2:

$$\text{Max } Z = x_1 + 2x_2 + x_3$$

Subject to:

$$x_1 - x_2 \quad > = 10$$
$$2x_1 + x_2 \quad = < 35$$
$$x_2 + x_3 = < 20$$



b^T = (10, 35, 20)

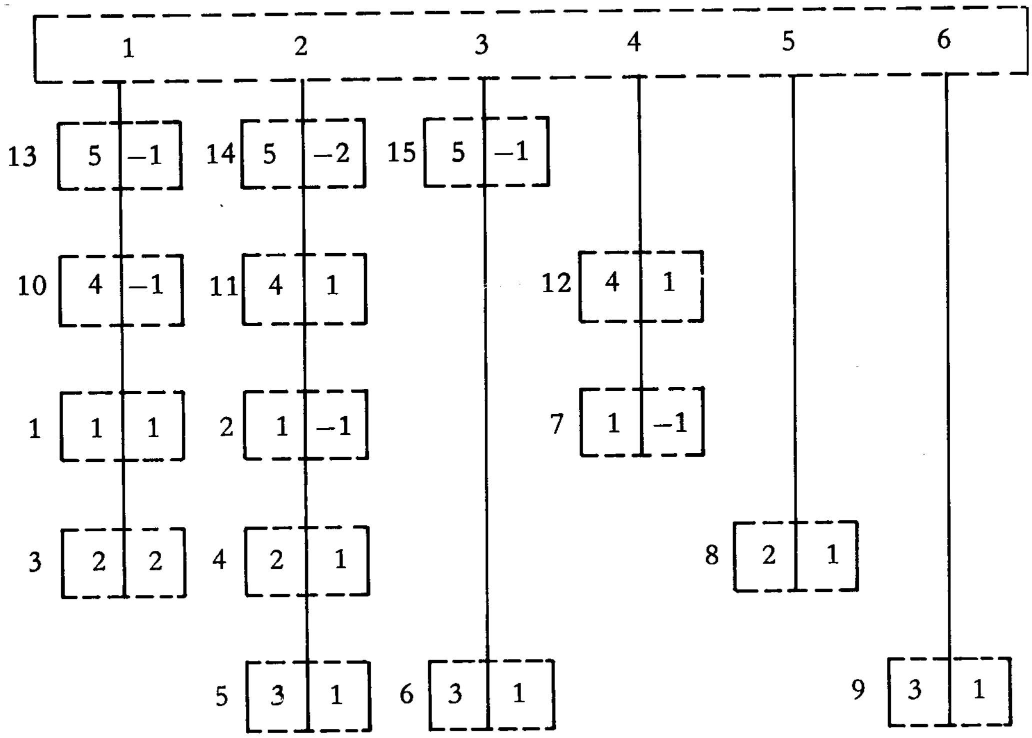
4-a. arrow and box representation

i	V(i)	P(i)	1(i)
1	1	1	3
2	–1	1	4
3	2	2	0
4	1	2	5
5	1	3	0
6	1	3	0

b^T = (10, 35, 20)

4-b vector representation

Figure 4. Chained vector representation of main problem.



$b^T = (10, 35, 20, -10, 0)$, NBT = (0, 5, 6, 0, 0)

5-a. arrow and box representation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
V(i)	1	−1	2	1	1	1	−1	1	1	−1	1	1	−1	−2	−1
P(i)	1	1	2	2	3	3	1	2	3	4	4	4	5	5	5
L(i)	3	4	0	5	0	0	0	0	0	1	2	7	10	11	6

$b^T = (10, 35, 20, -10, 0)$, $NB^T = (0, 5, 6, 0, 0)$

5-b vector representation

Figure 5. Chain vector representation of standard form.

NB is a vector where its i^{th} entry is equal to j if x_j is the i^{th} basic variable, and is equal to zero if the i^{th} basic variable is an artificial variable.

The problem obtained in this way is solved by the second algorithm (R. S. M.) The utilization of the chained vector also reduces the arithmetic operations per iteration, especially when the coefficient matrix contains a large number of zero entries. The number of multiplications (division is considered as a multiplication) and additions (subtraction is considered as an addition) per iteration of both R. M. S. and C. V. S. are compared and is given in Table 2.

Table 2. Comparison of R. M. S. and C. V. S.

Method		Operation		
		Pivoting	$z_j - c_j$'s	Total
R. M. S.	Mult.	$(m+1)^2$	$m(n-m)$	$m(n-m) + (m+1)^2$
	Add.	$m(m+1)$	$m(n-m)$	$m(n+1)$
C. V. S.	Mult.	$(m+1)^2$	$\sum_{j \in R} n_j$	$\sum_{j \in R} n_j + (m+1)^2$
	Add.	$m(m+1)$	$\sum_{j \in R} n_j$	$\sum_{j \in R} n_j + m(m+1)$

n_j is the number of nodes in the chained vector of the j^{th} column of the coefficient matrix in standard form. R is the set of non-basic vectors.

CONCLUSIONS

A procedure has been developed to efficiently solve the linear programming problems whose matrix of coefficients contains a large number of "zero" entries. The modified Chained Vector Simplex (C. V. S.) algorithm, was introduced as an alternative to Simplex and Revised Simplex Methods. It was demonstrated that the C. V. S. could be very useful when the matrix of coefficients contains a large number of "zero" entries in a linear, programming problem. By utilization of this algorithm, there will be no main memory problems and the CPU time is reduced considerably. Finally, comparison of C.V. S. and R. S. M. has been carried out and reduction of operation needed for multiplication (division) and addition (subtraction) were demonstrated.

This algorithm is applied to simulate a real time automatic control of air traffic in Europe [9], where the goal was to calculation the takeoff times of the aircrafts as a linear programming problem. The results were very satisfactory from the view point of CPU time and efficient use of memory storage.

REFERENCES

1. JERROLD L. WAGENER, Principles of programming, John Wiley & Sons, Inc, 1980.
2. ROBERT J. RADER, Advanced software design techniques, Petrocelli Books, Inc, 1978.
3. WILLIAM R SMYTH & LYNWOOD A JOHNSON, Introduction to linear programming with applications, Prentice Hall Inc. 1966.
4. R. BOUDAREL, J. DELMAS & P. GUICHET, Command optimal des processus, Dunod, Paris, 1968.
5. M. G. SINGH & TITLI, Systems: Decomposition, optimization and control, Pergamon Press 1978.
6. MICHEL. SIMONARD, Programmation lineair, technique du calcul economique, Dunod (39), Paris, 1972.
7. SAUL I. GASS, Linear programming, methods and applications, McGraw-Hill Book Company, 1969.
8. MOKHTAR S. BAZARRA & JOHN J. JARVIS, Linear programming and network flows, John Wiley & Sons, Inc, 1977.
9. S. KHANMOHAMMADI, Gestion automatisee du trafic aerien sur un ensemble d'aeroprts, Sup Aero, Toulouse, France, 1983.