Report n. 39

A LP CODE IMPLEMENTATION

Giuliano Gasparotto

Pisa, 1991

This research is partially supported by the Ministery of University and Scientific and Technological Research (MURST).

A LP CODE IMPLEMENTATION

G.Gasparotto

Preface

Many economic problems such as resources allocations or portfolio selection or goods loading or stock cutting may be formulated as Linear Fractional problems. The scientific interest for such a kind of economic problems has increased as well as the research in Fractional Programming field.

Previous studies of our group have given many theoretical results and, as an effect, the complete draft or the simple modification of algorithms able to solve problems of this type.

But the aim of these studies was also to extend the results of the investigations, started by Wagner-Yuau in 1968, to the better known L F P algorithms in order to identify, by means of those theoretical results, the most efficient ones and to make a comparison among them.

Thus we needed for that a ductile L P code preferably working on a P C which have, at the moment, an extended utilization both in the education and in the research. Existing products running on large computers, in spite of their reliability, did not give any possibility to be modified. Within this reason we have started in writing an, *ad hoc* , Linear Programming Code.

Our next step will be the realization of a Revised Simplex Method code to be used in a Branch and Bound procedure to attempt at the Integer Linear Programming.

1 Introduction

This report contains a brief description of an implemented primal simplex algorithm able to solve a problem of the type:

$$\begin{aligned} &\text{Minimize or Maximize} \quad \mathbf{cx} \\ &\text{Subject to} \quad \mathbf{Ax} \lesseqgtr \mathbf{b} \\ &\qquad \mathbf{x} \geq \mathbf{0} \\ &\text{with } \mathbf{b} \lesseqgtr 0 . \end{aligned}$$

in the attempt to give the reader a real possibility to use this code, if not to modify it.

For any theoretical description of the algorithm we refer the reader to specialized texts.

2 Equipment Needs

This program can be run on an Apple personal computer.

The minimum required hardware equipment is represented by a Macintosh SE standard configuration with an optional Printer.

The minimum required software includes a Microsoft FORTRAN 2.4 Compiler for the Apple Macintosh.

3 Program Description

3.0 First Level Description : MAIN PROGRAM

A main program calls the following subprograms:

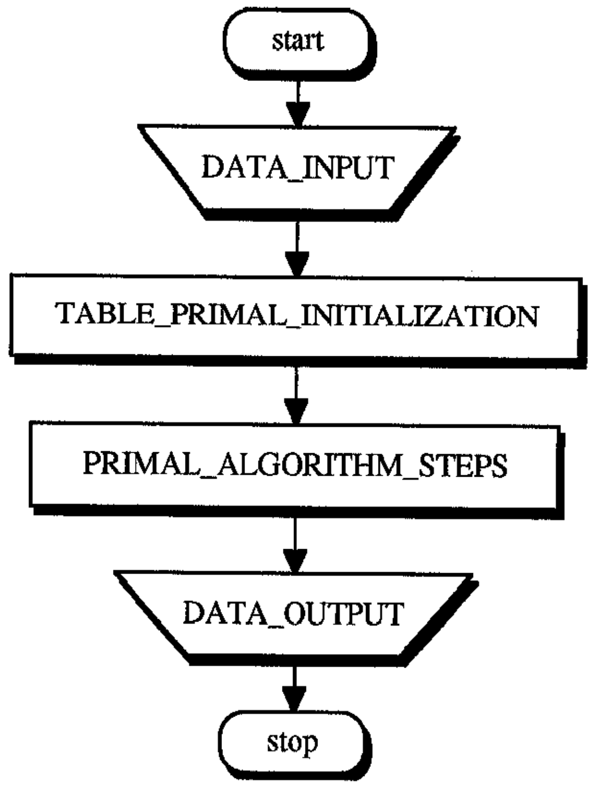
DATA_INPUT which reads input data from an existing file LP-DATA-in.

TABLE_PRIMAL_INITIALIZATION which initializes the problem and creates the first simplex tableau.

PRIMAL_ALGORITHM_STEPS which is at the head of the algorithm's iterations.

DATA_OUTPUT which creates a file, LP-DATA-out, where output data are written for further elaborations.

First level scheme : MAIN PROGRAM



Remark-3.0-1: before and after PRIMAL_ALGORITHM_STEPS the subprogram TABLEAU_PRINTING(_) creates two differents files, LP-TABLEAU-initial (1) and LP-TABLEAU-final (3), where initial and final simplex tableaus are written.

3.1 DATA_INPUT

This subprogram reads problem's input data.

For the following problem:

$$\max f(x) = c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 + c_5x_5 + c_0$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + a_{25}x_5 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 + a_{35}x_5 \geq b_3 \end{cases}$$

$$x_i \geq 0 \ , \forall i \in \{1,...,5\}$$

the schema of the input data is the following:

```
DATA of PROBLEM:
problem of MINimization (1) or MAXimization (-1)
-1
number M of the constraints (< or = 10)
3
number N of the variables (< or = 30)
5
objective function value:
c0
coefficients of the objective function:
c1      c2      c3      c4      c5
row of the N-coefficients of the M-constraints:
a11     a12     a13     a14     a15
a21     a22     a23     a24     a25
a31     a32     a33     a34     a35
type of the M-constraints: LE (-1) , EQ (0) , GE (1)
-1      0       1
right hand side of the M-constraints:
b1      b2      b3
END of DATA
```

Remark-3.1- 1: a problem of minimum or maximum can be proposed.

Remark-3.1- 2: the maximum number of constraints/variables is less then or equal to ten/thirty. These figures can be increased, within memory limitations, by means of suitable modifications [see **A Appendix**].

Remark-3.1- 3: the objective-function's value must be specified, because of the possibility to use the output as an input. When no c_0 is present in the problem its value must, any way, set to zero.

Remark-3.1- 4: different data on the same row may be separated by a key "TAB" or, at least, a blank character. The data may be written on differents rows. A data equivalent to zero must be explicitly written.

3.2 TABLE_PRIMAL_INITIALIZATION

This subprogram creates first simplex tableau,

a) trasforming any problem of maximization in a problem of minimization, because the algorithm's implementation operates only in such a case.

b) trasforming any constraint with a negative right-hand-side in a constraint with a positive one, changing its sign.

c) looking for a basis. In this order

1) it adds all necessary slacks variables. If not sufficient

2) it checks for an effective variable already basic and modifies consequently the objective function, still keeping in the original problem. If not sufficient

3) it adds all necessary auxiliary variables and creates the auxiliary objective function, going over to the auxiliary problem.

3.3.0 Second Level Description : PRIMAL_ALGORITHM_STEPS

This subprogram, core of the implementation, supports the functions of the algorithm:

a) by means of the subprograms

ENTERING_VARIABLE_STEP, which chooses the variable "entering in the basis", by the criterion

$$\{ x_k : c_k = \min c_j , \ c_j < 0 \},$$

SORTING_VARIABLE_STEP, which chooses the variable "sorting of the basis", by the criterion

$$\{ \ x_i \ : \ b_i / a_{ik} = \min \ b_j / a_{jk} \ , \ \ a_{jk} > 0 \ \},$$

TABLE_ITERATION-STEP, which operates the tableau iteration, according to the primal simplex algorithm's method.

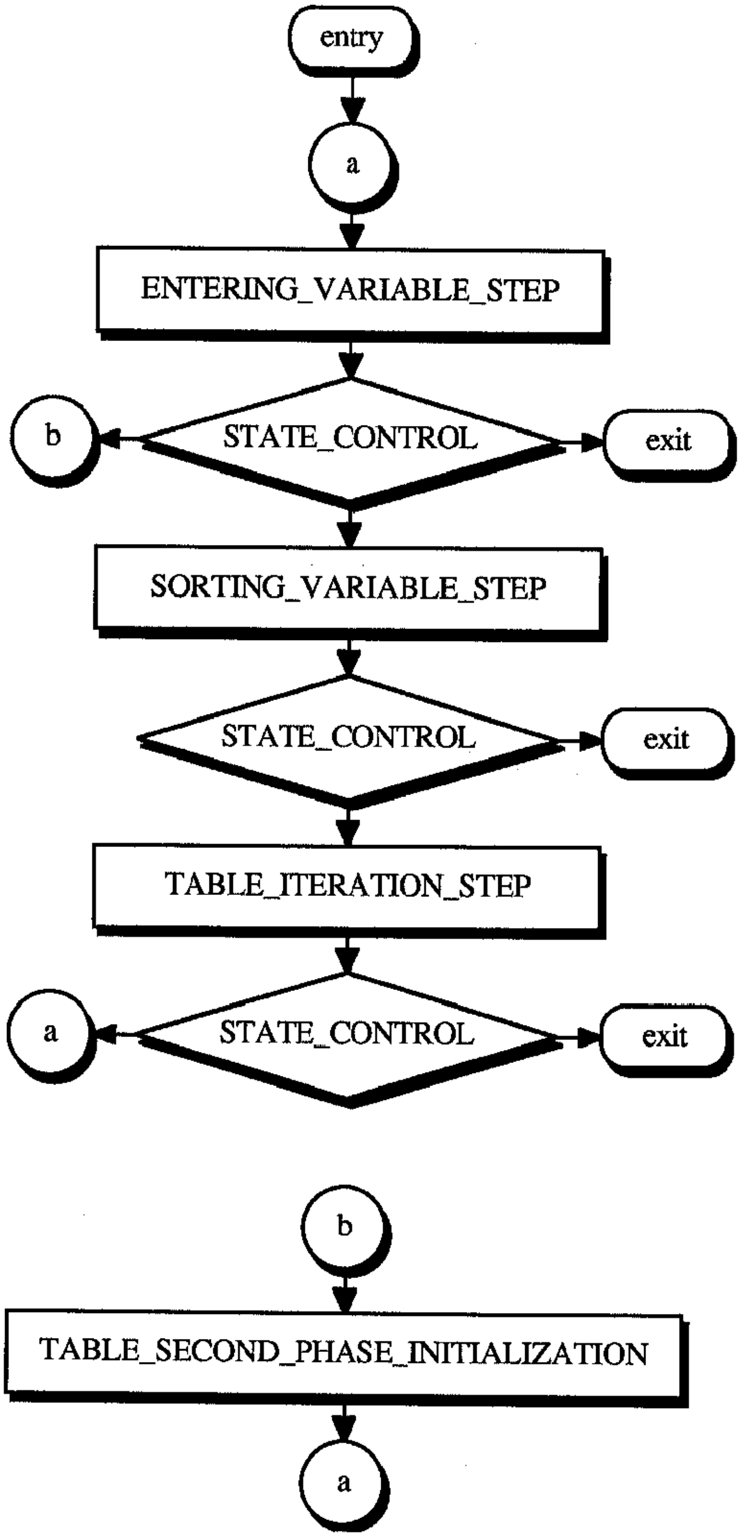
Remark-3.3.0 - 1: a degeneration control will be operated at this level.
Remark-3.3.0 - 2: at this level a control on the maximum number of iterations is also performed.

b) by means of the subprogram

TABLE_SECOND_PHASE_INITIALIZATION, which initializes again the problem, by the subprogram TABLE_ITERATION_STEP, in order to apply the second phase, if an auxiliary problem has been created in the TABLE_PRIMAL_INITIALIZATION subprogram and an optimal solution at level zero has been found in the first phase.

Remark-3.3.0 - 3: any redundant constraint will be eliminated.

Second level scheme : PRIMAL ALGORITHM STEPS



Remark-3.3.0 - 4: the subprogram TABLEAU_PRINTING(2) , if called after TABLE_ITERATION_STEP and TABLE_SECOND_PHASE_INITIALIZATION , will write in the LP_TABLEAU_current file, created by MAIN PROGRAM, the current simplex tableau.

STATE_CONTROL subprogram is at the head of the stages of the algorithm.

3.3.1 STATE_CONTROL

This subprogram recognizes differents situation, depending on the algorithm's phase, and sends all suitable messages on the display unit to the user.

So, if called after ENTERING_VARIABLE_STEP subprogram:

- $\exists\, c_k < 0$: first & second phase → **SORTING_VARIABLE_STEP**
- $\bar{\exists}\, c_k < 0$
  - first phase
    - o.f. > 0 : "feasible region is empty" → < return >
    - o.f. = 0 : "ending of first phase, beginning of second phase" → < b >
  - second phase
    - $\exists\, c_k = 0$: "problem has one or more optimal solutions" < return >
    - $\bar{\exists}\, c_k = 0$: "problem has one optimal solution" → < return >

if called after SORTING_VARIABLE_STEP subprogram:

- $\exists\, a_{jk} > 0$
  - $b_j = 0$: < cycle control > → **TABLE_ITERATION_STEP**
  - $b_j \neq 0$ → **TABLE_ITERATION_STEP**
- $\bar{\exists}\, a_{jk} > 0$: "problem has no solution: the region is unbounded" → < return >

if called after TABLE_ITERATION_STEP subprogram:

- $b_j = 0$: < cycle control >
  - problem presents a cycle case" → < return >
  - otherwise → ENTERING_VARIABLE_STEP
- $b_j \neq 0$ → ENTERING_VARIABLE_STEP

3.4 DATA_OUTPUT

This subprogram creates a file, where data results are written in the same way as the input data. Thus, at this point, a new elaboration is possible if either a maximum number of iterations has been achieved or the user wants to operate a cut on the feasible region, after a normal stop has been reached, by adding one or more constraints.

Remark-3.4.1: if the maximum number of iterations has been achieved during first phase, some cautions have to be taken because only the auxiliary objective function will be output, but no difference between effective and slack variables, on one side, and no auxiliary variables, on the other, will be kept.

4 Examples

4.1 Problem - 1

$$\max f = 3x_1 + 2x_2 + x_3 + 5x_4$$

$$\begin{cases} 3x_1 + 4x_2 + 5x_3 + 6x_4 \leq 5 \\ 2x_1 + 6x_2 + x_3 + 5x_4 \leq 6 \\ x_1 + x_2 + 5x_3 + x_4 = 7 \end{cases}$$

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

The LP-DATA-in file will be the following:

```
DATA of PROBLEM :
problem of MINimization (1) or MAXimization (-1)
-1
number M of the constraints ( < or = 10 )
3
number N of the variables ( < or = 10 )
4
objective function value:
0.0
coefficients of the objective function:
3        2        1        5
row of the N-coefficients of the M-constraints:
3        4        5        6
2        6        1        5
1        1        5        1
type of the M-constraints:  LE (-1) , EQ (0) , GE (1)
-1       1        0
right hand side of the M-constraints:
5        6        7
END of DATA
```

Running the program on the display will appear the messages:

```
  beginning of the phase : 1

 ENTERING_VARIABLE : k_ENTER,COST =  2, -7.00000
 SORTING_VARIABLE : i_row,i_cnstrnt,i_SORT,PIVOT = 2, 2, 7, 6.00000

 ENTERING_VARIABLE : k_ENTER,COST =  3, -4.83333
 SORTING_VARIABLE : i_row,i_cnstrnt,i_SORT,PIVOT = 1, 1, 5, 4.33333

 ENTERING_VARIABLE : k_ENTER,COST =  0, .000000

  ending of the phase : 1

THE FEASIBLE REGION OF THE PROBLEM IS EMPTY ***

  total number of iterations equal : 2
```

which signals too that this problem is characterized by an empty region. At this time four files are generated.

The LP-TABLEAU-initial file is omitted for this example.
In the LP-TABLEAU-current file all the simplex tableaux are memorized, at every iteration. Loking in this file it is possible to follow all the phases of the algorithm signalized on the display:
i.e.by cost equal -7.000, the effective variable x2 = 2e, in column 2, which will enter in the basis; by pivot equal 6.000, the auxiliary variable x7 = 7a, in the second constraint in row 2 and column 7, which will sort of the basis and so on.

* * * CURRENT TABLEAU AT ITERATION : 0 * * *

MINIMUM-OBJECTIVE-FUNCTION-VALUE = 13.0000

- VALUE	COSTS:	1c	2c	3c	4c	5c	6c	7c	8c
.0000	O.F.	3.0000	2.0000	1.0000	5.0000	.0000	.0000	.0000	.0000
-13.0000	A.F.	-3.0000	-7.0000	-6.0000	-6.0000	.0000	1.0000	.0000	.0000

b	FUNCTION: VARIABLES:	1e	2e	3e	4e	5s	6s	7a	8a
5.0000	EQ	3.0000	4.0000	5.0000	6.0000	1.0000	.0000	.0000	.0000
6.0000	EQ	2.0000	6.0000	1.0000	5.0000	.0000	-1.0000	1.0000	.0000
7.0000	EQ	1.0000	1.0000	5.0000	1.0000	.0000	.0000	.0000	1.0000

* * * CURRENT TABLEAU AT ITERATION : 1 * * *

MINIMUM-OBJECTIVE-FUNCTION-VALUE = 6.00000

- VALUE	COSTS:	1c	2c	3c	4c	5c	6c	7c	8c
-2.0000	O.F.	2.3333	.0000	.6667	3.3333	.0000	.3333	-.3333	.0000
-6.0000	A.F.	-.6667	.0000	-4.8333	-.1667	.0000	-.1667	1.1667	.0000

b	FUNCTION: VARIABLES:	1e	2e	3e	4e	5s	6s	7a	8a
1.0000	EQ	1.6667	.0000	4.3333	2.6667	1.0000	.6667	-.6667	.0000
1.0000	EQ	.3333	1.0000	.1667	.8333	.0000	-.1667	.1667	.0000
6.0000	EQ	.6667	.0000	4.8333	.1667	.0000	.1667	-.1667	1.0000

* * * CURRENT TABLEAU AT ITERATION : 2 * * *

MINIMUM-OBJECTIVE-FUNCTION-VALUE = 4.88462

- VALUE	COSTS:	1c	2c	3c	4c	5c	6c	7c	8c
-2.1538	O.F.	2.0769	.0000	.0000	2.9231	-.1538	.2308	-.2308	.0000
-4.8846	A.F.	1.1923	.0000	.0000	2.8077	1.1154	.5769	.4231	.0000

b	FUNCTION: VARIABLES:	1e	2e	3e	4e	5s	6s	7a	8a
.2308	EQ	.3846	.0000	1.0000	.6154	.2308	.1538	-.1538	.0000
.9615	EQ	.2692	1.0000	.0000	.7308	-.0385	-.1923	.1923	.0000
4.8846	EQ	-1.1923	.0000	.0000	-2.8077	-1.1154	-.5769	.5769	1.0000

The LP-TABLEAU-final file and the LP-DATA-out file are omitted for this example.

4.2 Problem - 2

$$\min f = -3x_1 - 2x_2$$

$$\begin{cases} -x_1 & \leq 9 \\ x_1 + x_2 & \geq 7 \\ 2x_1 & = 8 \end{cases}$$

$$x_1, x_2 \geq 0$$

The LP-DATA-in file will be the following:

```
DATA of PROBLEM :
problem of MINimization (1) or MAXimization (-1)
1
number M of the constraints less or equal 10
3
number N of the variables less or equal 10
2
objective function value:
0
coefficients of the objective function:
-3      -2
row of the N-coefficients of the M-constraints:
-1      0
1       1
2       0
type of the M-constraints:  LE (-1) , EQ (0) , GE (1)
-1      1       0
right hand side of the M-constraints:
9       7       8
END of DATA
```

On the display will appear the messages:

```
  beginning of the phase : 1

 ENTERING_VARIABLE : k_ENTER,COST =  1, -2.00000
 SORTING_VARIABLE : i_row,i_cnstrnt,i_SORT,PIVOT = 3, 3, 5, 2.00000

 ENTERING_VARIABLE : k_ENTER,COST =  0, .000000

  ending of the phase : 1

  beginning of the phase : 2

 ENTERING_VARIABLE : k_ENTER,COST =  4, -2.00000
 SORTING_VARIABLE : i_row,i_cnstrnt,i_SORT,PIVOT = 0, 0, 0, .000000

THE PROBLEM HAS NO SOLUTION: THE REGION IS UNBOUNDED ***

  total number of iterations equal : 1
```

which signals that the problem is characterized by an unbounded region.

The LP-TABLEAU-initial and the LP-TABLEAU-final files, where the only first and last simplex tableaux are memorized, are the following:

* * * * * INITIAL TABLEAU * * * * *

MINIMUM-OBJECTIVE-FUNCTION-VALUE = 8.00000

5c	- VALUE	COSTS:	1c	2c	3c	4c
.0000	14.0000	O.F.	-1.0000	.0000	.0000	-2.0000
.0000	-8.0000	A.F.	-2.0000	.0000	.0000	.0000

5a	b	FUNCTION: VARIABLES:	1e	2e	3s	4s
.0000	9.0000	EQ	-1.0000	.0000	1.0000	.0000
.0000	7.0000	EQ	1.0000	1.0000	.0000	-1.0000
1.0000	8.0000	EQ	2.0000	.0000	.0000	.0000

* * * * * FINAL TABLEAU * * * * *

MINIMUM-OBJECTIV-FUNCTION-VALUE = -18.0000

- VALUE	COSTS:	1c	2c	3c	4c
18.0000	O.F.	.0000	.0000	.0000	-2.0000

b	FUNCTION: VARIABLES:	1e	2e	3s	4s
13.0000	EQ	.0000	.0000	1.0000	.0000
3.0000	EQ	.0000	1.0000	.0000	-1.0000
4.0000	EQ	1.0000	.0000	.0000	.0000

Neither LP-TABLEAU-current, nor LP-DATA-out files are reported for this example.

4.3 Problem - 3

$$\min f = x_2$$

$$\begin{cases} x_1 + 2x_2 \geq 2 \\ x_1 - \frac{1}{2} x_2 \leq 2 \end{cases}$$

$$x_1, x_2 \geq 0$$

The running of the code will generate on the display the sequence of messages :

```
 beginning of the phase : 1

ENTERING_VARIABLE : k_ENTER,COST =  2, -2.00000
SORTING_VARIABLE : i_row,i_cnstrnt,i_SORT,PIVOT = 1, 1, 5, 2.00000

ENTERING_VARIABLE : k_ENTER,COST =  0, .000000

 ending of the phase : 1

 beginning of the phase : 2

ENTERING_VARIABLE : k_ENTER,COST =  1, -.500000
SORTING_VARIABLE : i_row,i_cnstrnt,i_SORT,PIVOT = 1, 1, 2, .500000

ENTERING_VARIABLE : k_ENTER,COST =  0, .000000

THE PROBLEM HAS ONE OPTIMAL SOLUTION ***

 total number of iterations equal : 2
```

which signals that the problem has a solution. Following in LP-TABLEAU-current file all the informations given, it is possible to see the passage fron the first to the second phase.

```
      * * *  CURRENT TABLEAU AT ITERATION : 0  * * *

MINIMUM-OBJECTIVE-FUNCTION-VALUE = 2.00000

  - VALUE     COSTS:        1c          2c          3c          4c
5c


       .0000  O.F.        .0000      1.0000       .0000       .0000
.0000

    -2.0000   A.F.      -1.0000     -2.0000      1.0000       .0000
.0000


            FUNCTION:
    b     VARIABLES:        1e          2e          3s          4s
5a

      2.0000    EQ       1.0000      2.0000     -1.0000       .0000
1.0000

      2.0000    EQ       1.0000      -.5000       .0000      1.0000
.0000
```

* * * CURRENT TABLEAU AT ITERATION : 1 * * *

MINIMUM-OBJECTIVE-FUNCTION-VALUE = .000000

- VALUE	COSTS:	1c	2c	3c	4c	5c
-1.0000	O.F.	-.5000	.0000	.5000	.0000	-.5000
.0000	A.F.	.0000	.0000	.0000	.0000	1.0000
b	FUNCTION: VARIABLES:	1e	2e	3s	4s	5a
1.0000	EQ	.5000	1.0000	-.5000	.0000	.5000
2.5000	EQ	1.2500	.0000	-.2500	1.0000	.2500

* * * CURRENT TABLEAU AT ITERATION : 1 * * *

MINIMUM-OBJECTIV-FUNCTION-VALUE = 1.00000

- VALUE	COSTS:	1c	2c	3c	4c
-1.0000	O.F.	-.5000	.0000	.5000	.0000
b	FUNCTION: VARIABLES:	1e	2e	3s	4s
1.0000	EQ	.5000	1.0000	-.5000	.0000
2.5000	EQ	1.2500	.0000	-.2500	1.0000

* * * CURRENT TABLEAU AT ITERATION : 2 * * *

MINIMUM-OBJECTIV-FUNCTION-VALUE = .000000

- VALUE	COSTS:	1c	2c	3c	4c
.0000	O.F.	.0000	1.0000	.0000	.0000
b	FUNCTION: VARIABLES:	1e	2e	3s	4s
2.0000	EQ	1.0000	2.0000	-1.0000	.0000
.0000	EQ	.0000	-2.5000	1.0000	1.0000

The LP-TABLEAU-final file is omitted.

In the LP-DATA-out file data results of the last simplex tableau are memorized, as data input schema, as follow:

```
DATA of PROBLEM :
problem of MINimization (1) or MAXimization (-1)
1
number M of the constraints less or equal 10
2
number N of the variables less or equal 10
4
objective function value:
.000000
coefficients of the objective function:
     .0000      1.0000       .0000       .0000
row of the N-coefficients of the M-constraints:
    1.0000      2.0000     -1.0000       .0000
     .0000     -2.5000      1.0000      1.0000
type of the M-constraints: EQ (0)
0 0
right hand side of the M-constraints:
    2.0000       .0000
END of DATA
```

A Appendix : The code uses four common blocks of memory area.

A.1 In TYPE_PROBLEM are memorized informations, used only for the printing, about variables, constraints and the problem . Informations are only alphabetical.

VARIABLE_type	"e" if effective, "s" if slack, "a" if auxiliary.
CONSTRAINT_type	"LE" if less or equal, "EQ" if equal, "GE" if greater or equal.
PROBLEM_type	"MAX" for a problem of maximization, "MIN" for a problem of minimization

A.2 In INITIAL_PROBLEM the input data are memorized in the subprogram DATA_INPUT and updated in the subprogram TABLE_PRIMAL_INITIALIZATION. Informations are only numeric.

OPT_initial	-1 for a maximization problem, 1 for a minimization problem.
OBJ_O_initial	stores the initial value of the objective function for the original problem and can be updated only if any effective variable, with non zero cost, is introduced in the basis.
OBJ_A_initial	stores the initial value of the objective function for the auxiliary problem.
M_initial	store the number of the constraints.
N_initial	stores the number of the variables and is updated depending on the numbers of variables introduced, slacks or auxiliaries.
C_O_initial	stores the coefficients of the objective function for the original problem and can be transformed only if any effective variable, with non zero cost, is introduced in the basis. It is a vector of dimension 30.
C_A_initial	stores the coefficients of the objective function for the auxiliary problem. It is a vector of dimension 30.
A_initial	stores the coefficients of the matrix and is updated depending on the number of the variables introduced, slacks or auxiliaries. It is a matrix of dimension 10 x 30.
B_initial	stores the right-hand-side of the constraints. It is a vector of dimension 30.
SIGN_initial	-1,0,1 for each constraint of the type LE, EQ, GE respectively. It is a vector of dimension 10.

A.3 In CURRENT_PROBLEM all data of INITIAL_PROBLEM are memorized in the subprogram TABLE_PRIMAL_INITIALIZATION and updated during the elaboration in subprogram PRIMAL_ITERATION_STEPS, more exactly in subprogram TABLE_ITERATION_STEP or TABLE_SECOND_PHASE_INITIALI - ZATION. Informations are only numeric.

A.4 In SERVICE_PROBLEM all informations needed for the elaboration are stored. Informations are only numeric.

nmbr_E	stores the number of effective variables.
nmbr_S	stores the number of slack variables.
nmbr_A	stores the number of auxiliary variables.
nmbr_E_B	stores the number of effective variables which are basic.
nmbr_S_B	stores the number of slack variables which are basic.
nmbr_A_B	stores the number of auxiliary variables which are basic.
nmbr_E_nt_B	stores the number of effective variables which are not basic.
nmbr_S_nt_B	stores the number of slack variables which are not basic.
nmbr_A_nt_B	stores the number of auxiliary variables which are not basic.
active_CNSTRNT_vector	is a vector in which any component stores an index of an active constraint. It is initialized in the subprogram TABLE_PRIMAL_INITIALIZATION and is updated, if that is the case, in subprogram TABLE_SECOND_PHASE_INITIALIZA TION. It is a vecor of dimension 10.
vector_CNSTRNT_related_BSC_VRBL	is a vector in which any component, related to a constraint, stores the index of the variable basic in that constraint. It is a vecor of dimension 10.
nt_BSC_VRBLS_vector	is a vector in which any component, related to a variable, is set to -1 or 1 depending whether the variable is basic or not. It is a vecor of dimension 30.

Remark-A : The maximum number of effective, slack and auxiliary variables, all together, is 30, the maximum number of constraints is 10.

References

[1] A.Carnes and W.W.Cooper - *Management Models and Industrial Applications of Linear Programming*. Volumes I/II, John Wiley & Sons,Inc., 1967.

[2] H.M.Wagner and J.S.C.Yuan - "Algorithm Equivalence in Linear Fractional Programming".*Management Science* (1968), Vol 14, pp. 301-306.

[3] G. Hadley - *Linear Programming*. Addison-Wesley Publishing Company, 1969.

[4] A.H.Land and S.Powel - *Fortran Codes for Mathematical Programming: Linear, Quadratic and Discrete*. John Wiley & Sons, 1973.

[5] Mokhtar S.Bazaraa and John J.Jarvis - *Linear Programming and Network Flows*. John Wiley & Sons, 1977.

[6] G.R.Birtran "Experiments with Linear Fractional Problems". *Naval Research Logistics Quarterly* (1979), Vol. 26, pp. 689-693.

[7] Warren J.Erikson and Owen P.Hall,jr - *Computer Model for Management Science*. Addison-Wesley Publishing Company, 1983.

[8] A.Cambini and L.Martein - *Introduzione all'Algebra Lineare - Elementi di Programmazione Lineare e Non Lineare*. Libreria Scientifica G.Pellegrini - Pisa, 1984.

[9] Microsoft Corporation - *Microsoft FORTRAN Compiler for the Apple Macintosh*. Microsoft Corporation, 1985.

[10] G.Gasparotto - *On the Charnes-Cooper Transformation in Linear Fractional Programming* Report n 3 (1987), Dipartimento di Statistica e Matematica Applicata all'Economia, University of Pisa, Italy.