AN ALGORITHM FOR LARGE LINEAR DYNAMIC OPTIMIZATION WITH APPLICATION TO GENERATION SCHEDULING

A. Friedlander
C. Lyra
H. Tavares
and
E. L. Medina

RELATÓRIO TÉCNICO Nº 27/87

ABSTRACT. This paper proposes a method to solve large linear dynamic problems (L.D.P.).
L. D. P. constraint matrices have a particular structure, called staircase. The algorithm presented achieves efficiently by skillful utilization of this structure. It is built upon the classic work by Bartels-Golub, which uses the L–U decomposition of the constraint matrix. A compromise between structure preservation and numerical stability is searched throughout the solution process.
The method was applied to a deterministic generation scheduling problem of the São Francisco River hydroeletric power system, in Northeastern Brazil. A summary of numerical experiences is presented and a discussion concerning the compromisse "preservation of structure" versus "stability" is carried out.

Universidade Estadual de Campinas
Instituto de Matemática, Estatística e Ciência da Computação
IMECC – UNICAMP
Caixa Postal 6065
13.081 - Campinas, SP
BRASIL

Julho – 1987

1. INTRODUCTION

Sparse matrices with staircase structure arise in Linear Dynamic Programming problems.

Krivonozhko and Chebotarev (1976) and Propoi and Krivonozhko (1978) developed an algorithm to solve a Linear Dynamic Problem. This algorithm adapted the Simplex Method to this particular problem and used the Forrest-Tomlin (1972) updating scheme, which has good properties related with sparsity preservation but is not numerically stable.

Fourer (1979, 1982, 1983, 1984) surveys the properties of staircase matrices and the varieties of Gaussian elimination for staircase matrices.

In this paper we present an algorithm to solve the L.D.P. problem, which adapts the Simplex Method's steps to this particular structure. The updating scheme is based on Bartels - Golub's method (1969) and a compromise between the structure preservation and stability is established in the computations process.

2. STATEMENT OF THE PROBLEM

The Linear Dynamic Programming (L.D.P) problem has the form

$$\min J(x,u) = \sum_{t=0}^{T-1} [c(t+1)\,x(t+1) + d(t)\,u(t)]$$

subject to

$$x(t+1) = A(t)\,x(t) + B(t)\,u(t)$$

$$x(0) = x^{o}$$

$$C(t)\,x(t) + D(t)\,u(t) = f(t)$$

$$\alpha \leq x(t+1) \leq \beta$$

$$\gamma \leq u(t) < \delta$$

$$\text{for} \quad t = 0,1,\dots,T-1$$

where

$x(t)$: state vector $(n \times 1)$

u(t): control vector $(r \times 1)$

f(t): resource vector, given, $(m \times 1)$, $m \leq r$

$x(0) = x^o$: initial state vector

A(t), B(t), C(t), D(t): matrices with dimensions $(n \times n)$, $(n \times r)$, $(m \times n)$, $(m \times r)$, respectively.

$c(t+1)$, $d(t)$: cost vectors of dimensions $(1 \times n)$, $(1 \times r)$, respectively.

T: number of time periods, fixed.

The constraint matrix of this problem has the following form

$$
\begin{array}{c} \\ \\ \mathbb{A} = \end{array}
\begin{array}{ccccccc}
\overset{u(0)}{r} & \overset{x(1)}{n} & \overset{u(1)}{r} & \overset{x(2)}{n} \ldots & x(T-1) & u(T-1) & x(T) \\
\end{array}
$$

$$
\mathbb{A} = \begin{bmatrix}
D(0) & & & & & & \\
B(0) & -\mathbb{I} & & & & & \\
 & C(1) & D(1) & & & & \\
 & A(1) & B(1) & -\mathbb{I} & & & \\
 & & & \ddots & C(T-1) & D(T-1) & \\
 & & & & A(T-1) & B(T-1) & -\mathbb{I}
\end{bmatrix}
$$

fig. 1

where $\mathbb{I}$ is a $(n \times n)$, identity matrix.

3. BASIS FACTORIZATION

Any submatrix $\mathbb{B}$ of $\mathbb{A}$ satisfies

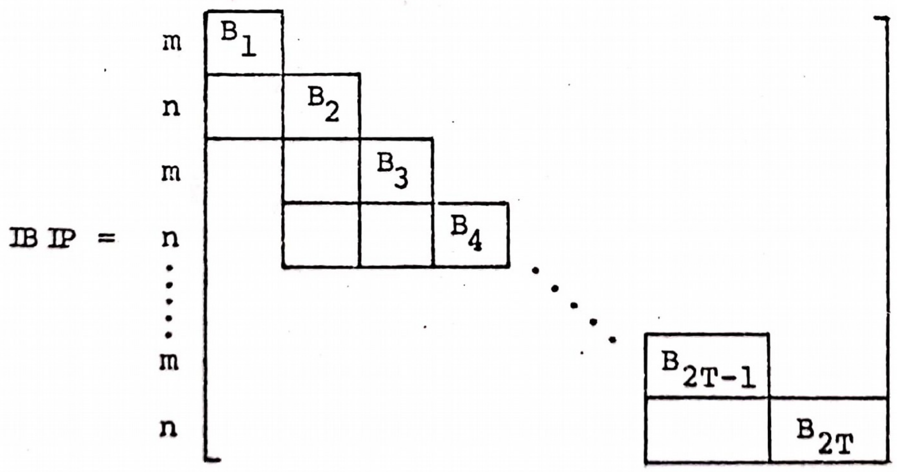

Fig. 2

for some permutation matrix P.

Let $\mathbb{B}$ be a basis, then the block B_1 must have m independent columns. We perform a stable row-pivoting within this (m × m) submatrix of block B_1. As the number of columns of B_1 can be greater than m, after this process we may have some columns of B_1, that remain unpivoted.

Let Δ(1) be the submatrix formed by the remaining columns of B_1, and the first m-rows. To continue the factorization, consider the submatrix whose columns are the remaining columns of B_1 and the columns of B_2, and whose rows are the first n-rows after the last pivoted row. Once again the fact the $\mathbb{B}$ is a basis guarantees the performance of a stable row-pivoting within this submatrix. We continue this process until the whole matrix $\mathbb{B}$ is factored. Finally we have

$$\mathbb{L}^{-1}\mathbb{B} = \mathbb{U}$$

where

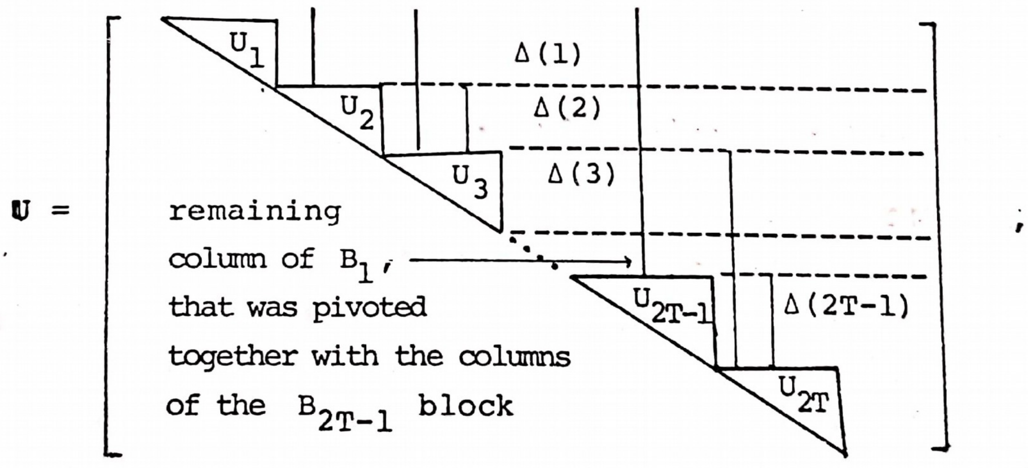

Fig. 3

$$\mathbb{L} = L_{[(m+n)T-1]}\ P_{(m+n)\cdot T-1} \cdots\cdots\ L_2P_2\ \ L_1P_1\ ,$$

the L_i are elementary matrices, and the P_i elementary permutation matrices.

$\Delta(i)$ is the matrix whose columns are the remaining ones and whose rows are the corresponding rows of U_i, $1 \leq i \leq 2T-1$. The whose process will not produce fill-in outside the staircase except for the remaining columns. Note that the columns of any U_i, $0 \leq i \leq 2T$ may represent state or control variables from any previous stage.

At each iteration of the Revised Simplex Method three linear systems of equations must be solved

i) determining the current basic solution.

ii) evaluating the updated column corresponding to the variable entering the basis.

iii) determining the simplex multipliers vector associated with the current basis $\mathbb{B}$

In the resolution of these systems we use the $\mathbb{L}\,\mathbb{U}$ factorization of $\mathbb{B}$, obtained by the process described above, and take account of the particular structure of $\mathbb{U}$ and of the inherited sparsity of the elementary L_i matrices whose product is $\mathbb{L}$..

4. UPDATING OF THE $\mathbb{L}\,\mathbb{U}$ FACTORIZATION

At each iteration of the Simplex Method a column is chosen to enter the basis, and a column is then determined to leave the basis. We shall make some considerations about the relative positions between these variables, before describing the updating process.

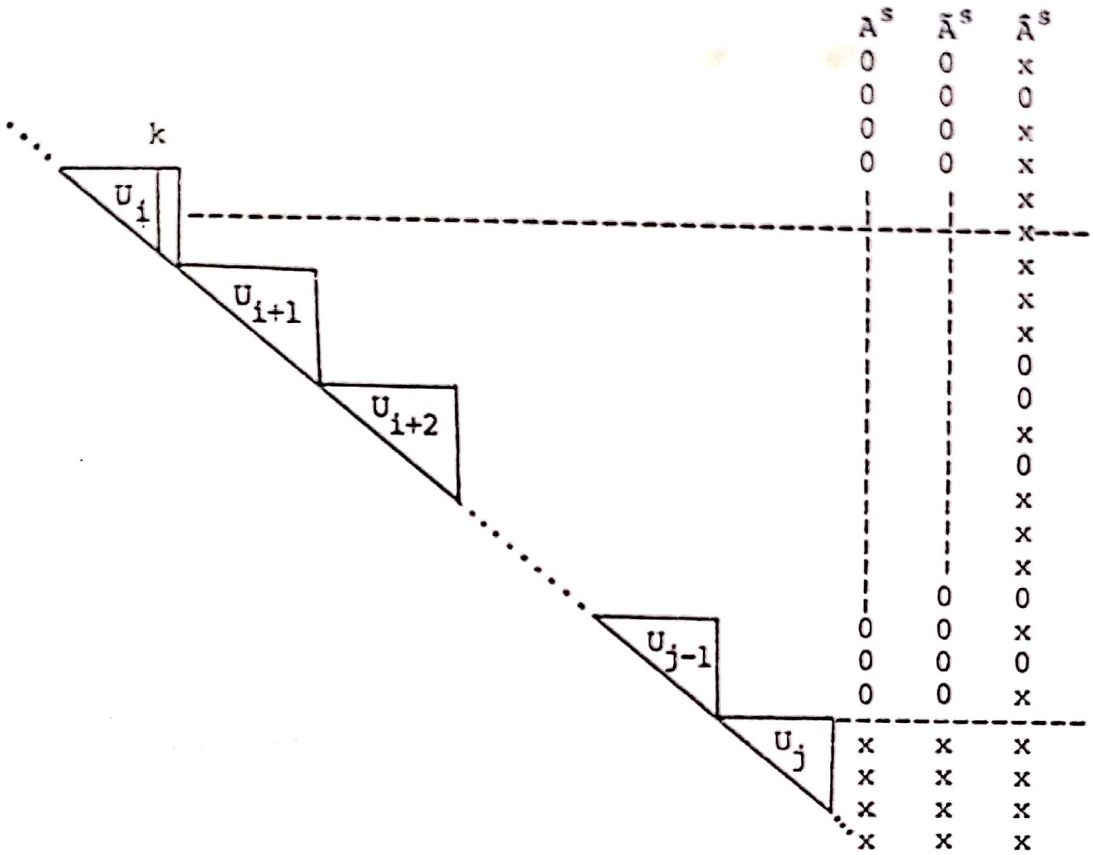

fig. 4

The variable leaving the basis is represented in figure 4 within U_i and the entering variable is associated with U_j, $j > i$. This association is made considering the stage in which the variable stays in A.

Let A^s be the column in A corresponding to the entering variable. Let $\tilde{A}^s = \mathbb{L}^{-1} A^s$

and
$$\mathbb{U}\, \hat{A}^s = \tilde{A}^s$$

Let k be the position of the leaving variable relative to U_i, A^s may have its first non-zero element on the first row of U_j, not above. The same is true for $\tilde{A}^s$. It is clear that the k-th component of $\hat{A}^s$ must be different from zero.

Let $(\hat{a}_{i1}, \ldots \hat{a}_{ik}, \ldots \hat{a}_{im})^t$ be the A^s components corresponding

to the rows of U_i (we suppose that the dimension of U_i is m).
As $U \hat{A}^s = \bar{A}^s$, we have

$$U_i \begin{pmatrix} \hat{a}_{i1} \\ \hat{a}_{ik} \\ \hat{a}_{im} \end{pmatrix} + \Delta(i)\hat{a}_k = 0$$

where $\hat{a}_k$ are the components of $\hat{A}^s$, that stay under the last row of U_i.

Then

$$\begin{pmatrix} \hat{a}_{i1} \\ \hat{a}_{ik} \\ \hat{a}_{im} \end{pmatrix} = U_i^{-1}\Delta(i)\hat{a}_k$$

and we conclude that the k-th row of $U_i^{-1}\Delta(i)$ must have a non-zero element.

This guarantees the existence of a column of $\Delta(i)$, linearly independent of the $(m - 1)$ column left in U_i when we retire its k-th column.

To describe the updating process, we have to distinguish the following two situations.

i) The leaving column stays in U_i, and the entering column is associated with U_j, with $j > i$.

Let k be the position of the leaving column relative to U_i, which we suppose of dimension m. We bring the columns $(k + 1, k + 2, \ldots, m)$ of U_i back to positions $(k, k + 1, \ldots, m - 1)$ respectively, and column k is put in place of the m-th column of U_i. This permutations transform the lower-triangular U_i in a Hessenberg matrix H_i. The H_i matrix is pivoted, in order to eliminate the sub-diagonal, choosing the element with largest absolute value between the diagonal and sub-diagonal in each column as pivot (Bartels-Golub, 1969).

These operations increase the number of elementary L_i that define $\mathbb{L}^{-1}$.

Clearly, these operations involve only the rows of $\mathbb{B}$, corresponding to U_i. The $\Delta(i)$ columns, which at the end of these eliminations have a zero in the m-th row of U_i, are linearly dependent of the first (m-1) columns of U_i.

We choose the first $\Delta(i)$ column which has a non-zero element on this row. The considerations made at the beginning of this section guarantee the viability of this choice.

We permute the chosen column of $\Delta(i)$ with the leaving column (which is now the last column of U_i). With this permutation, some elements are created under $\mathbb{U}$'s diagonal, and the next step is their elimination. Observe that the fact that the column introduced in U_i, was the first one with non-zero element in the last U_i's row, guarantees that this last elimination produces no fill-in on $\mathbb{U}$'s columns between the two permuted columns. Also alterations in $\mathbb{U}$ will only occur on the $\Delta(i)$ columns with non-zero element on the pivoting-row.

After this process the leaving column stays in a U_ℓ with $\ell > i$. We repeat this procedure until the leaving column reaches the position of the last U_j's column. At this moment we arrive to situation:

ii) The leaving column is in U_i and the entering column is associated with U_j, $j \leq i$. We proceed just as in situation i.

The difference is, that, in this case, it may happen that the whole $\Delta(i)$ row, in which we look for a non-zero element, is null if this is the case, the leaving column is deffinitely retired from $\mathbb{U}$ and replaced by the updated entering column.

Once again, this change of columns, creates elements below

U's diagonal, but the fact that the whole $\Delta(i)$'s pivoting row is null, guarantees that no fill-in is produced in U when those elements are eliminated. If $\Delta(i)$ has a non-zero element on the pivoting row we proceed as in i), repeating the process until for some $\ell \geq i$, $\Delta(\ell)$'s pivoting row is null or $\ell = 2T$. Then we may effectivize the change of columns. It is clear that the whole process preserves the staircase structure with exception of the "remaining columns" of the factorization, and some new $\Delta(i)$ type columns that way be introduced during the updating process. Also some of these $\Delta(i)$ columns may be deleted during this process.

5. APPLICATIONS TO A GENERATION SCHEDULING PROBLEM

The algorithm presented has been applied to the weekly generation scheduling problem of the São Francisco River hydroelectric power system, in Northeastern Brazil. This system, represented in fig 5, comprises two reservoir plants, Sobradinho and Moxotõ, and two run-of-river plants, Paulo Afonso (P.A.) I-III and P.A. IV, conected to the reservoir of Moxotõ by an artificial canal

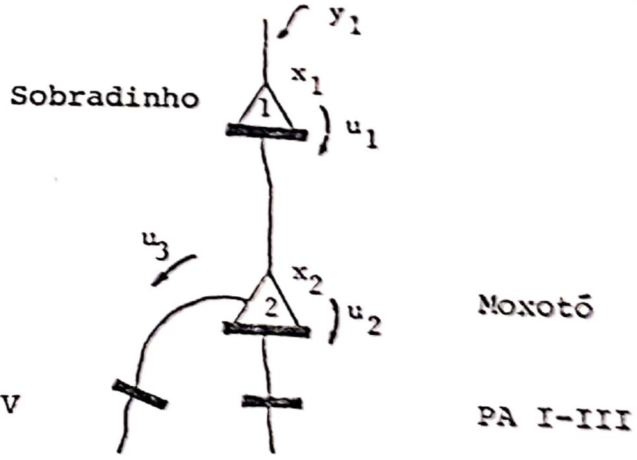

fig. 4

Plants dynamics and hydroelectric coupling are summarized in the following water conservation equations:

$$x_1^{t+1} = x_1^t + y_1^t - u_1^t - v_1^t \qquad (1)$$

$$x_2^{t+1} = x_2^t + y_2^t + u_1^{t-\tau} + v_1^{t-\tau} - u_2^t - u_3^t - v_2^t - v_3^t \qquad (2)$$

$$t = 0,1,\ldots,T-1$$

where,

x_1^t, x_2^t: volume of water stored at Sobradinho (plant 1) and Moxotó (plant 2), respectively, at the end of period t;

y_j^t : volume of independent water inflow forecasted to plant j at interval t (adjusted to include losses);

u_1^t : volume of water discharged from Sobradinho during interval t;

u_2^t : volume of water discharged from Moxotó and P.A. I-III ;

u_3^t : volume of water discharged from P.A. IV ;

v_1^t : spill of Sobradinho during interval t;

v_2^t : spill of Moxotó and P.A.-I-III ;

v_3^t : spill of P.A.-IV ;

τ : time lag for water displacement from Sobradinho to Moxotó ;

T : final stage

Contents of reservoirs have upper and lower bounds,

$$\underline{x}_j \leq x_j^t \leq \bar{x}_j \qquad j = 1,2 \quad t = 1,2\ldots T \qquad (3)$$

There are also local constraints on discharges and spills,

$$\underline{u}_j \leq u_j^t \leq \bar{u}_j \qquad (4)$$

$$0 \leq v_j^t \qquad (5)$$

$$j = 1,2,3$$

$$t = 0,1,\ldots, T-1$$

Energy balance is expressed by the following equations,

$$h_1^t + h_2^t + \Delta^t = d^t \qquad t = 0,\ldots,T-1 \tag{6}$$

$$h_1^t = g_1(x_1^t, u_1^t, v_1^t) \tag{7}$$

$$h_2^t = g_2(x_2^t, u_2^t) + g_3(u_2^t, v_2^t) + g_4(x_2^t, u_3^t) \tag{8}$$

where.

Δ^t : load shedding (or thermal generation)

d^t : energy demand

g_j : non-linear functions

The weekly generation scheduling problem for this system searches feasible controls (u_j^t, v_j^t) that enable meeting load requirements with the least utilization of energy reserves.

It has been found from operational data that linear approximations of functions g_j are accurate enough for the purpose of this problem. Thus, equations (7) and (8) may be expressed as

$$h_1^t = A_1 \cdot u_1^t + E_1 \cdot x_1^t + C_1 \tag{9}$$

$$h_2^t = A_2 \cdot u_2^t + A_3 \cdot u_3^t + E_2 \cdot x_2^t + C_2 \tag{10}$$

where A_j, E_j and C_j are constants.

A planning horizon of two weeks is considered, divided in 42 discrete intervals of 8 hours each. The time lag for water displacement is 3 days, which implies that $\tau = 9$ in equation (2).

The problem can be solved in two steps. The first step, which can be interpretated as a Phase I of the simplex procedure, searches to meet energy demand avoiding load shedding (or thermal generation). If this goal is accomplished, a second step searches a solution that maximizes water reserves at Sobradinho (the reserves at Moxotó are negligible).

The first step can be formally stated as

$$\text{Min} \sum_{t=0}^{41} \Delta^t$$

s.t. (1)-(6),(9),(10)

$$\Delta^t \geq 0$$

$$\left.\begin{array}{ll} x_1^o, x_2^o & \\ y_1^t, y_2^t & t = 0,\ldots.41 \\ u_1^{t-9} + v_1^{t-9} & t = 0,\ldots.8 \\ d^t & t = 0,\ldots.41 \end{array}\right\} \text{Known}$$

For the second step the objective function changes into

$$\text{Max} \quad x_1^{42}$$

and $\Delta^t (t = 0,\ldots.41)$ are dropped from the energy balance equations.

6.NUMERICAL EXPERIENCES

The first experience considers the following values for the constants in equations (9) and (10),

$A_1 = 7,8$

$E_1 = 0,0073$

$A_2 = 34,3$ $C_1 + C_2 = -354,7$

$E_2 = 0,1871$

$A_3 = 37,0$

These constants are expressed in standard unities as, for example, $10^6 m^3$ volume and average MW for energy.

It can be easily verified (Friedlander, 1986) that the submatrices corresponding to a given period are very ill-conditioned, with these constants as coefficients. As the compromise to preserve structure restricts permutations within these submatrices , this ill-conditioning may affect the stability.

Indeed, this case study led to an unstable process, detected by the appearance of basic solutions that violated bound constraints.

To overcome the unfeasibility we retrieved a previous feasible basic solution and, starting from this point, performed some iterations where the preservation of structure was relaxed. With this strategy the process converged to an optimal solution , confirming the hypothesis that the ill-conditioned submatrices caused the unstability.

Another experience considers a linear approximation for the generation functions g_j where dependence on storage is only implicity considered, in the estimations of A_j. The values obtained were

$A_1 = 8,26$

$E_1 = E_2 = 0$

$A_2 = 32,98$

$C_1 = C_2 = 0$

$A_3 = 34,39$

In this case the algorithm converged without any stability problems. An optimal solution for the first step was obtained with

195 iterations . Each iteration required an average of 1 second of C.P.U. Reserves of energy were maximized in the second step with 25 iterations that required an average of 0,3 seconds of C.P.U.

The program was coded in Fortran 77 and ran on a VAX/785, with VMS operating system.

6. CONCLUSION

This paper presented a new method to solve large linear dynamic problems which seeks a compromise between preservation of dynamical structure and numerical stability.

The algorithm was applied to a generation scheduling problem which illustrates aspects of its behaviour in a real dynamic problem.

REFERENCES:

Bartels, R.H., and Golub, G.H. (1969). The simplex method using LU decomposition, Comm. ACM 12, pag. 266-268

Friedlander, A., (1986) "Optimization with staircase structure in the constraints". (in portuguese). Tese de Doutorado - State University of Campinas - Brazil.

Forrest, J.J.H. and Tomlin, J.A. (1972) Updating triangular factors of the basis to mantain sparsity in the product form simplex method, Mathematical Programming, 2, pag. 263-278.

Fourer, R. (1979). Sparse Gaussian elimination of staircase linear systems, Technical Report Sol 79-17, Systems Optimization Laboratory, Dept of Operations Research, Stanford University.

Fourer, R. (1982) Solving staircase linear programs by the simplex method, 1: inversion, Mathematical Programming 23, pag 274-313.

Fourer, R. (1983) , Solving staircase linear programs by the simplex method, 2: Pricing, Mathematical Programing 25, pag 251-292

Fourer, R. (1984) Staircase Matrices and Systems , SIAM Review, vol. 26 , nº 1, pag 1-70

Krivonozhko, V.Z. and Chebotarev, S.P. (1976). Factorization method for linear dynamic programming problems developing systems. Plenum Publishing Corporation

Murtagh, B.A. and Saunders, M.A. (1978). Large-scale linearly constrained optimization, Mathematical Programming 14, pag 41-72

Propoi, A. and Krivonozhko, V.E. (1978). The simplex method for Dynamic Linear Programs - Report RR - 78 - 14 - int. IASA , Luxenburg , Austria