THE SIMPLEX METHOD FOR DYNAMIC LINEAR PROGRAMS

A. Propoi and V. Krivonozhko

RR-78-14
September 1978

International Institute for Applied Systems Analysis
A-2361 Laxenburg, Austria

Printed by NOVOGRAPHIC
Maurer-Lange-Gasse 64
1238 Wien

PREFACE

Many problems of interest to applied IIASA projects can be formulated within the framework of dynamic linear programming (DLP). Examples are long-range energy, water, and other resources supply models, problems of national settlement planning, long-range agriculture investment projects, manpower and educational planning models, resources allocation for health care, etc.

There are many different approaches and methods for tackling DLP problems, which use decomposition, penalty functions, augmented Lagrangian, nondifferentiable optimization technique, etc.

This paper presents an extension of the simplex method, the basic method for solution of dynamic linear programming problems. The paper consists of three parts. Part I, "dual systems of DLP", concerns theoretical properties of the problem, primarily, duality relations; Part II, "the dynamic simplex method: general approach" describes the idea and the theory of the method; and Part III, "a basis factorization approach", gives a complete description of the algorithm, as well as the connection with the basis factorization approach. Part III also includes a numerical example that is not trivial for a general LP algorithm but is solved very easily by using the dynamic simplex method. Part II is written in a language more familiar to control theory specialists, Part III is closer to linear programming. All parts are written as independent papers with their own references and thus can be read independently. However, the whole paper comprises a theory of finite-step methods for DLP. The next development of the research might be first numerical tests on the behavior of the method and thus a judgment of its efficiency, and second, extensions of the approach to other classes of structured linear programming (for example, to DLP problems of the transportation type).

The paper has its origin in previous IIASA publications. These and other related papers in DLP are listed at the end of this report.

The authors would like to express their acknowledgment to George B. Dantzig for his long-standing interest in and support for this work. Many thanks also to to Etienne Loute for his valuable comments and suggestions.

SUMMARY

There are two major approaches in the finite-step methods of structured linear programming: decomposition methods, which are based on the Dantzig-Wolfe decomposition principle, and basis factorization methods, which may be viewed as special instances of the simplex method.

In this paper, the second approach is used for one of the most important classes of structured linear programming–dynamic linear programming (DLP).

The paper presents a finite-step method for DLP–the dynamic simplex method. This is a natural and straightforward extension of one of the most effective static LP methods–the simplex method–for DLP. A new concept –a set of local bases (for each time step)–is introduced, thus enabling considerable reduction in the computer core memory requirements and CPU time.

The paper is in three parts. Part I, "dual systems of DLP" concerns theoretical properties of the problem; it is written by A. Propoi. The pair of dual problems are formulated and the relations between them are established, which allows us to obtain optimality conditions, including the maximum principle for primal and the minimal principle for dual problems. The results are formulated for a canonical form of DLP, and then modifications and particular cases are considered.

Part II, "the dynamic simplex method: general approach" and Part III "a basis factorization approach", written by V. Krivonozhko and A. Propoi, give the description of the dynamic simplex method and its extensions.

In Part II construction of a set of local bases and their relation to the conventional "global" basis in LP are given. A special control variation and the corresponding objective function variation as applied to this set of local bases are described. This part is written in a language more familiar to control theory specialists.

Part III describes the separate procedures of the dynamic simplex method: primal solution, dual solution, pricing, updating and the general scheme of

the algorithm. The connection between the method and the basis factorization approach is also shown. A numerical example and a theoretical evaluation of the algorithms reveal the efficacy of the approach. The extensions of the method (dual and primal-dual versions of the algorithms, application to DLP problems with time lags) are briefly discussed in the final part of the paper. This part is closer to LP specialists.

CONTENTS

I. DUAL SYSTEMS OF DYNAMIC LINEAR PROGRAMMING

1. Introduction

The impact of linear programming (LP) [1,2] models and methods in the practice of decision making is well known. However, because of computational difficulties in its solution, its application has been for the most part one-staged and static in nature; that is to say, the problem of the best allocation of limited resources is usually considered at some fixed stage in the development of a system.

When the system to be optimized is developing (not only in time, but possibly in space as well), a one-stage approach is inadequate. In this case, decisions are scheduled over time and the problem of optimization becomes a dynamic, multi-stage one.

In fact, almost every static LP model has its own dynamic variant, the latter being of growing importance because of the increasing role of planning in decision making [3].

Within the context of dynamic linear programming (DLP), new problems arise. For the static LP, the basic question consists of determining the optimal program. For the dynamic case, the questions of feedback control, stability and sensitivity are also important. Hence, the DLP theory and methods should be based both on the methods of linear programming and on the methods of control theory, Pontryagin's maximum principle [4] and its discrete version [5] in particular.

The aim of this paper is a presentation of theoretical properties of dynamic linear programs, especially duality relations and optimality conditions. The pair of dual problems are formulated and the relations between them are obtained. From these relations, optimality conditions (including maximum principle for primal and minimum principle for dual problems) are derived. The results are formulated for a canonical form of DLP, then modifications of the canonical form are given.

We consider the DLP problem in the following canonical form:

Problem 1. To find a control $u = \{u(0), u(1), \ldots, u(T-1)\}$ and a trajectory $x = \{x(0), x(1), \ldots, x(T)\}$, satisfying the state equations

$$x(t+1) = A(t)x(t) + B(t)u(t) + s(t) \qquad (t = 0,1,\ldots,T-1) \tag{1}$$

with initial state

$$x(0) = x^0 \tag{2}$$

and constraints

$$G(t)x(t) + D(t)u(t) \leq f(t) \tag{3}$$

$$u(t) \geq 0 \qquad (t = 0,1,\ldots,T-1) \tag{4}$$

which maximize the objective function (performance index)

$$J_1(u) = a(T)x(t) + \sum_{t=0}^{T-1} (a(t)x(t) + b(t)u(t)) \quad . \tag{5}$$

Here the vector $x(t) = \{x_1(t), \ldots, x_n(t)\}$ defines the state of the system at stage t in the state space X, which is supposed to be the n-dimension euclidean space E^n, the vector $u(t) = \{u_1(t), \ldots, u_r(t)\} \in E^r$ specifies the controlling action at stage t; the vector $s(t) = \{s_1(t), \ldots, s_n(t)\}$ defines the external effects on the system (uncontrolled, but known a priori in the deterministic models). Vectors $f(t) \in E^m$, x^0, $s(t)$, $a(t)$, $b(t)$ and the matrices $A(t)$, $B(t)$, $G(t)$, $D(t)$ with conforming dimensions are given.

In the vector products the right vector is a column and the left vector is a row; thus, ab is the inner product of vectors a and b.

The choice of a canonical form of the problem is to some extent arbitrary making various modifications and particular cases of Problem 1 possible. In the last section, some examples will be considered; however, it should be noted that such modifications can be either reduced to Problem 1 or it is possible to use the results stated below for Problem 1 [5] for the modifications. We conclude this section with some definitions.

Definition. A *feasible control* of Problem 1 is a vector sequence $u = \{u(0),\ldots,u(T-1)\}$ which satisfies with the *trajectory* $x = \{x(0),\ldots,x(T)\}$ all constraints (1) to (4). An *optimal control* is a feasible control u^*, which maximizes (5). Feasible control and the trajectory constitute *feasible process* $\{u,x\}$.

2. Duality Relations

Note that if $T = 1$, Problem 1 becomes the conventional LP problem. On the other hand, Problem 1 itself can be considered as one "large" LP problem, with constraints on its variables in the form of equalities (1), (2) and inequalities (3), (4).

Let us introduce the Lagrange function for Problem 1:

$$L(u,x;\lambda,p) = a(T)x(T) + \sum_{t=0}^{T-1} (a(t)x(t) + b(t)u(t)) \tag{6}$$

$$+ \sum_{t=0}^{T-1} p(t+1)\,(A(t)x(t) + B(t)u(t) + s(t) - x(t+1))$$

$$+ \sum_{t=0}^{T-1} \lambda(t)\,(f(t) - G(t)x(t) - D(t)u(t))$$

$$+ p(0)(x^0 - x(0)) \quad .$$

In the above $p(t) \in E^n (t=T,\ldots,0)$, $\lambda(t) \in E^m$, $\lambda_i(t) \geq 0$ ($i = 1, \ldots,m$; $t = T-1,\ldots,0$) are the Lagrange multipliers for the constraints (1), (2) and (3) respectively.

Employing the Lagrange function (6), the following subproblems are now considered [6]:

$$\sup_{x;u\geq 0} \inf_{p;\lambda\geq 0} L(u,x;\lambda,p) = \omega_1 \tag{7}$$

$$\inf_{p;\lambda\geq 0} \sup_{x;u\geq 0} L(u,x;\lambda,p) = \omega_2 \quad . \tag{8}$$

The problems (7), (8) will be studied separately. It is assumed that an optimal process (solution) of the original Problem 1 exists and is denoted by $\{u^*,x^*\}$.

Lemma 2.1. Any solution $\{u^*,x^*\}$ *of Problem 1 is also a solution of (7); moreover the objective performance index (5) satisfies*

$$J_1(u^*) = \omega_1 \quad .$$

If $\omega_1 > -\infty$, *then any solution of (7) is a solution of Problem 1; otherwise the system of constraints (1)-(4) is inconsistent.*

The proof, being a standard one in mathematical programming, is omitted here.

Now let us rewrite the Lagrange function in the "dual" form:

$$L(\lambda,p;u,x) = (a(T) - p(T))x(T) + \sum_{t=T-1}^{0} (p(t+1)A(t) - \lambda(t)G(t)$$

$$+ a(t) - p(t))x(t) + \sum_{t=T-1}^{0} (p(t+1)B(t) - \lambda(t)D(t)$$

$$+ b(t))u(t) + \sum_{t=T-1}^{0} (p(t+1)s(t) + \lambda(t)f(t)) + p(0)x^0$$

and consider the following dual problem.

Problem 2. To find a dual control $\lambda = \{\lambda(T-1),\ldots,\lambda(0)\}$ and a dual trajectory $p = \{p(T),\ldots,p(0)\}$ such that they satisfy the costate equations

$$p(t) = p(t+1)A(t) - \lambda(t)G(t) + a(t) \quad (t=0,\ldots,0) \tag{9}$$

with the boundary conditions

$$p(T) = a(T) \tag{10}$$

and constraints

$$-p(t+1)B(t) + \lambda(t)D(t) \geq b(t) \quad , \tag{11}$$

$$\lambda(t) \geq 0 \quad , \tag{12}$$

which minimize the dual performance index

$$J_2(\lambda) = p(0)x^0 + \sum_{t=0}^{T-1} (p(t+1)x(t) + \lambda(t)f(t)) \quad . \tag{13}$$

We shall call Problems 1 and 2 a pair of dual problems. It should be noted that dual Problem 2, as well as primal Problem 1, is a control problem, in which the variable $\lambda(t)$ specifies the dual controlling action at the stage t, the variable $p(t)$ is the dual state (costate) at the stage t; in the dual problem, time is taken in the reversed direction: $t = T-1,\ldots,1,0$.

So the following definitions are natural: The vector sequence $\lambda = \{\lambda(T-1),\ldots,\lambda(0)\}$ is a *dual control*; the corresponding sequence $p = \{p(T),\ldots,p(0)\}$, which is obtained from the dual state equations (9) with boundary condition (10), is a *dual (conjugate) trajectory*; the process $\{\lambda,p\}$, which satisfies all constraints (9) to (12) of Problem 2, is *feasible*. The feasible process $\{\lambda^*,p^*\}$, which minimizes (13), is *optimal (solution* of Problem 2).

The following proposition is proved in a similar manner to Lemma 2.1.

Lemma 2.2. Any solution $\{\lambda^*,p^*\}$ of Problem 2 is also a solution of the Problem (8) with $J_2(\lambda^*) = \omega_2$. If $\omega_2 < \infty$, then any solution of (8) is a solution of Problem 2; otherwise the system of constraints (9)-(12) is inconsistent.

Now we shall consider the relations between the dual Problems 1 and 2. First of all, the following assertion directly results from Lemmas 2.1 and 2.2.

Theorem 2.1. *For any feasible controls* u *and* λ *of the primal and dual Problems 1 and 2, the inequality*

$$J_1(u) \leq J_2(\lambda)$$

holds, where the values of $J_1(u)$ *and* $J_2(\lambda)$ *are computed from (5) and (13), using (1), (2) and (9), (10).*

For optimal controls u* and λ^*, the inequality of Theorem 2.1 becomes an equality.

Lemma 2.3. *(cf. [7]). The necessary conduction that* $\{u^* \geq 0, x^*\}$ *and* $\{\lambda^* \geq 0, p^*\}$ *be the optimal processes for the dual Problems 1 and 2 is that* $\{u^*, x^*; \lambda^*, p^*\}$ *be a saddle point for the Lagrange function (6), that is*

$$L(u^*,x^*;\lambda,p) \geq L(u^*,x^*;\lambda^*,p^*) \geq L(u,x;\lambda^*,p^*) \quad .$$

If $\{u^*,x^*\}$ *and* $\{\lambda^*,p^*\}$ *are optimal, then* $L(u^*,x^*;\lambda^*,p^*)$ *is the optimal value of the performance indices of dual Problems 1 and 2.*

Theorem 2.2. *(Duality Theorem). If one of the dual Problems 1 and 2 has an optimal control, then the other has an optimal control as well and the associated values of the performance indices of the primal and dual Problems 1 and 2 are equal:*

$$J_1(u^*) = J_2(\lambda^*) \quad .$$

If the performance index either of Problem 1 or 2 is unbounded (for Problem 1 from above and for Problem 2 from below), then the other problem has no feasible control.

The proof of Theorem 2.2 can be obtained in many ways. In particular, one can apply the duality theory of "static" LP [1,7], to Problem 1, regarding it as a static LP problem with constraints on the variable $u(t)$ and $x(t)$, both in the form of equalities (1), (2) and inequalities (3), (4), or using the dynamic programming approach one can reduce Problem 1 to a recurrence sequence of static linear programming problems and apply to them successively the LP duality theorem.

From the basic dual Theorem 2.2, the optimality and existence conditions follow for Problems 1 and 2:

Theorem 2.3. A feasible control u^* *is optimal if and only if there is a feasible* λ^* *with* $J_2(\lambda^*) = J_1(u^*)$. *A feasible control* λ^* *is optimal if and only if there is a feasible primary control* u^* *with* $J_1(u^*) = J_2(\lambda^*)$.

Theorem 2.4. (Existence Theorem). A necessary and sufficient condition that one (and thus both) of the dual Problems 1 and 2 have optimal controls is that both have feasible controls.

The above theorems are derived from their static analogues [1,7]. As such they represent nothing new. But in the dynamic case, the duality relations for each step t, which are stated below, are more interesting because they suggest, in a sense, a decomposition of the problem.

3. Optimality Conditions

Let us introduce the Hamilton functions

$$H_1(p(t+1),u(t)) = b(t)u(t) + p(t+1)B(t)u(t) \qquad (14)$$

for the primary Problem 1 and

$$H_2(x(t),\lambda(t)) = \lambda(t)f(t) - \lambda(t)G(t)x(t) \qquad (15)$$

for the dual Problem 2.

Lemma 3.1. For any controls u *and* λ *the following equality*

$$J_1(u) - J_2(\lambda) = \sum_{t=0}^{T-1} [H_1(p(t+1),u(t)) - H_2(x(t),\lambda(t))]$$

is valid.

Proof. Let us consider the difference

$$J_1 - J_2 = a(T)x(T) + \sum_{t=0}^{T-1} (a(t)x(t) + b(t)u(t))$$

$$- \sum_{t=0}^{T-1} (p(t+1)s(t) + \lambda(t)f(t)) - p(0)x^0 \quad .$$

Substituting the value $x(t)$, defined by the primary system (1), when $t = T-1$, and using the definition of the dual system (9), one can obtain

$$J_1 - J_2 = a(T)(A(T-1)x(T-1) + B(T-1)u(T-1) + s(T-1)) + a(T-1)x(T-1)$$

$$+ b(T-1)u(T-1) + \sum_{t=0}^{T-2} (a(t)x(t) + b(t)u(t)) - p(T)s(T-1)$$

$$- \lambda(T-1)f(T-1) + \sum_{t=0}^{T-2} (p(t+1)s(t) + \lambda(t)f(t)) - p(0)x^0$$

$$= H_1(p(T),u(T-1)) - H_2(x(T-1),\lambda(T-1) + p(T-1)x(T-1)$$

$$+ \sum_{t=0}^{T-2} (a(t)x(t) + b(t)u(t)) - \sum_{t=0}^{T-2} (p(t+1)s(t) + \lambda(t)f(t))$$

$$- p(0)x^0 = \ldots = \sum_{t=0}^{T-1} [H_1(p(t+1),u(t)) - H_2(x(t),\lambda(t))] \quad .$$

•

In Section 2 the relations were established between the objective functions of the primal and dual problems, which characterize the problem as a whole. Now "local" duality theorems will be obtained establishing relations between the Hamilton functions. For simplicity of statements, it is assumed that

Problem 1 (and, hence, Problem 2) has an optimal feasible solution.

Lemma 3.2. For any feasible process $\{u,x\}$ *and* $\{\lambda,p\}$ *the following inequalities hold:*

$$H_1(p(t+1),u(t)) \leq H_2(x(t),\lambda(t)) \qquad (t = 0,\dots,T-1) \quad .$$

Proof. One can obtain successively from (14),(12),(2),(15), (11), and (3):

$$H_1(p(t+1),u(t)) = b(t)u(t) + p(t+1)B(t)u(t) \leq b(t)u(t)$$

$$+ p(t+1)B(t)u(t) + \lambda(t)(f(t) - G(t)x(t) - D(t)u(t))$$

$$= H_2(x(t),\lambda(t)) + (p(t+1)B(t) - \lambda(t)D(t) + b(t))u(t)$$

$$\leq H_2(x(t),\lambda(t) \quad .$$

It should be noted that the statement of Theorem 2.1 also follows from Lemmas 3.1 and 3.2 for any feasible processes $\{u,x\}$ and $\{\lambda,p\}$.

Theorem 3.1. ("local" duality Theorem). For any feasible processes $\{u^*,x^*\}$ *of the primal and* $\{\lambda^*,p^*\}$ *of the dual to be optimal it is necessary and sufficient that the values of Hamilton functions are equal:*

$$H_1(p^*(t+1),u^*(t)) = H_2(x^*(t),\lambda^*(t)) \quad (t = 0,\dots,T-1) \quad .$$

Proof. One obtains from duality Theorem 2.2 and Lemma 3.1, that for optimal processes of dual Problems 1 and 2 the equality

$$\sum_{t=0}^{T-1} H_1(p^*(t+1),u^*(t)) = \sum_{t=0}^{T-1} H_2(x^*(t),\lambda^*(t)) \tag{16}$$

is valid. Hence from Lemma 3.2, it follows that the values of the Hamiltonians must be equal for $t = 0,1,\dots,T-1$ in case of optimal processes $\{u^*,x^*\}$ and $\{\lambda^*,p^*\}$.

Indeed, let us assume that it is not so, that is, let for some $0 \leq t \leq T-1$:

$$H_1(p^*(t+1), u^*(t)) \leq H_2(x^*(t), \lambda^*(t)) \quad .$$

This, however, is inconsistent with the equality (16). The contradiction completes the proof of the Theorem.

Considering the proof of Lemma 3.2 and the equality (16), it is not difficult to obtain that for optimality of $\{u^*, x^*\}$ and $\{\lambda^*, p^*\}$, it is necessary and sufficient that the following conditions be satisfied ($t = 0, \ldots, T-1$):

$$\lambda^*(t)[f(t) - G(t)x^*(t) - D(t)u^*(t)] = 0$$

$$[p^*(t+1)B(t) - \lambda^*(t)D(t) + b(t)]u^*(t) = 0 \quad .$$

From the above equalities and the definitions of dual constraints [7], one can obtain in the usual way the following "differential" (complementary) optimality conditions for Problems 1 and 2 (cf. [1,7]).

Lemma 3.3. *If both Problems 1 and 2 have feasible controls, then they have optimal controls* u^*, λ^*, *such that:*

if u^* *satisfies a constraint as an equation, then* λ^* *satisfies the dual constraint as a strict inequality;*

if λ^* *satisfies a constraint as an equation, then* u^* *satisfies the dual constraint as a strict inequality.*

Lemma 3.4. *If both Problems 1 and 2 are feasible then for any* i *either* $[G(t)x^*(t) + D(t)u^*(t)]_i < f_i(t)$ *for some optimal* u^* *and* $\lambda_i^*(t) = 0$ *for every optimal* λ^*; *or* $[G(t)x^*(t) + D(t)u^*(t)]_i = f_i(t)$ *for every optimal* u^* *and* $\lambda_i^*(t) > 0$ *for some optimal* λ^*.

For any j *either* $[-p^*(t+1)B(t) + \lambda^*(t)F(t)]_j > b_j(t)$ *for some optimal* λ^* *and* $u_j^*(t) = 0$ *for every optimal* u^*; *or*

$[-p^*(t+1)B(t) + \lambda^*(t)D(t)]_j = b_j(t)$ *for every optimal* λ^* *and* $u_j^*(t) > 0$ *for some optimal* u^*.

The conditions stated in Lemmas 3.3 and 3.4 are similar to the complementary slackness relations in linear programming [1,7]. From these lemmas, the known Kuhn-Tucker optimality conditions easily follow for Problems 1 and 2. As the assertions of the lemmas are not only necessary but also sufficient, it is not difficult to see that in order to investigate a pair of dual dynamic Problems 1 and 2 it is sufficient to consider a pair of dual "local" (static) problems of linear programming:

$$\max H_1(p(t+1),u(t))$$

$$G(t)x(t) + D(t)u(t) \leq f(t) \qquad u(t) \geq 0 \tag{17}$$

and

$$\min H_2(x(t),\lambda(t))$$

$$-p(t+1)B(t) + \lambda(t)D(t) \geq b(t) \qquad \lambda(t) \geq 0 \tag{18}$$

linked by the primary (1) and dual (9) state equations with boundary conditions (2) and (10).

So, any of the "static" duality relations or LP optimality conditions [1,2,7] for the pair of the dual LP problems (17) and (18) linked by the state equations (1), (2) and (9), (10) determine the corresponding optimality conditions for the pair of the dual DLP Problems 1 and 2. Such conditions have been formulated above; in a similar manner the following important optimality conditions are obtained.

Theorem 3.2. *(Maximum principle for primal Problem 1). For a control* u^* *to be optimal in the primal Problem 1, it is necessary and sufficient that there exists a feasible process* $\{\lambda^*,p^*\}$ *of the dual Problem 2, such that for* $t = 0,1,\ldots,T-1$ *the equality*

$$\max H_1(p^*(t+1),u(t)) = H_1(p^*(t+1),u^*(t))$$

holds, where the maximum is taken over all $u(t)$*, satisfying the constraints (3), (4) and* $\lambda^*(t)$ *is the optimal dual variable in the LP problem (18).*

Theorem 3.3. (Minimum principle for dual Problem 2). For a control λ^* *to be optimal in the dual Problem 2 it is necessary and sufficient that there exists a feasible process* $\{u^*,x^*\}$ *of the primal Problem 1, such that for* $t = 0,1,\ldots,T-1$ *the equality*

$$\min H_2(x^*(t),\lambda(t)) = H_2(x^*(t),\lambda^*(t))$$

holds, where the minimum is taken over all $\lambda(t)$*, satisfying the constraints (11), (12) and* $u^*(t)$ *is the optimal primary variable in the LP problem (17).*

These theorems can also be obtained by using the corresponding optimality conditions for discrete control systems [5].

4. Examples

In this section, duality relations for some typical examples will be given.

1. Problem 4.1. (without constraints on the state variables) We consider Problem 1 for which constraints are given only on the control variables.

$$\begin{aligned} & D(t)u(t) \leq f(t) \\ & u(t) \geq 0 \qquad (t = 0,1,\ldots,T-1) \quad . \end{aligned} \tag{19}$$

This is a special case of a DLP problem which reduces to T static LP problems

$$\max_{u(t)} H_1(p(t+1),u(t))$$

subject to constraints (19), where Hamilton function H_1 is defined from (14) and dual state variables $p(t+1)$ are directly computed from

$$p(t) = p(t+1)A(t) + a(t) \quad ,$$

$$p(t) = a(t) \qquad (t = T-1,\dots,1,0) \quad .$$

So, the dual controls $\{\lambda(t)\}$ are not used in this case.

2. <u>*Problem 4.3*</u>*. (with given left and right ends for the trajectory)* In Problem 1 let both ends of a trajectory be fixed:

$$x(0) = x^0 \quad , \qquad x(t) = x^T \quad .$$

In this case, the boundary conditions (10) for dual Problem 2 should be replaced by

$$p(T) + \lambda(T) = a(t)$$

and the term $\lambda(T)x^T$ added to the dual performance index (13).

3. <u>*Problem 4.3*</u>*. (with summary constraints)* Let the constraints for Problem 1 be given in the form

$$\sum_{t=0}^{T-1} [G(t)x(t) + D(t)u(t)] \le f \quad , \qquad u(t) \ge 0 \quad . \tag{20}$$

Although this case can be reduced to the canonical form of Problem 1 by introducing a new set of state equations [5], it is interesting to formulate the dual problem directly for the case (20). Only one dual control variable, λ, must be introduced here. Thus, the state equations (9) are replaced by

$$p(t) = p(t+1)A(t) - \lambda G(t)$$

with boundary conditions (10) and constraints

$$-p(t+1)B(t) + \lambda D(t) \ge 0 \quad , \qquad \lambda \ge 0 \quad .$$

The dual performance index becomes

$$J_2(\lambda) = p(0)x^0 + \lambda f$$

and the Hamilton function

$$H_2(\lambda) = \lambda f - \lambda \sum_{t=0}^{T-1} G(t)x(t) \quad .$$

The coincidence conditions of the Hamilton function's values (Theorem 3.1) in this case are:

$$\sum_{t=0}^{T-1} H_1(p(t+1),u(t)) = H_2(\lambda) \quad .$$

4. Problem 4.4. (with time delays) Problems with time delays are very important as they arise in many practical cases. We consider here the problem with delays on control variables:

$$x(t+1) = A(t)x(t) + \sum_{j=1}^{\mu} B(t-m_j)\,u(t-m_j) \tag{21}$$

$$(t = 0,1,\dots,T-1)$$

with given initial conditions:

$$x(0) = x^0 \tag{22}$$

$$u(-m_\mu) = u^0(-m_\mu),\dots,u(-m_1) = u^0(-m_1) \tag{23}$$

where $\{m_1,\dots,m_\mu\}$ is some given ordered set of integers.

The constraints on variable are supposed to be given in the form (3), (4) and performance index by

$$J_1(u) = a(T)x(T)$$

where $T > m_\mu$.

The dual problem for this case will be as follows.

To find a dual control $\lambda = \lambda\{(T-1-m_1),\dots,\lambda(T-1-m_\mu),$ $\lambda(T-1),\dots,\lambda(0),\dots,\lambda(-m_\mu)\}$ and a corresponding trajectory

$p = \{p(T), \ldots, p(0)\}$, satisfying the state equations (9) with (10) and constraints

$$\sum_{j=1}^{\mu} p(t+1+m_j)B(t) - \lambda(t)D(t) \leq 0$$

$$(0 \leq t \leq T-1-m_1)$$

$$p(1)B(-m_j) - \lambda(-m_j)D(-m_j) \leq 0 \qquad (j = 1, \ldots, \mu)$$

which minimize the performance index

$$J_2(\lambda) = p(0)x^0 + \sum_{j=1}^{\mu} \lambda(-m_j)u^0(-m_j) \quad .$$

The DLP problems with time delays on state variables can be considered in a similar way (cf. [5]).

5. Conclusion

The duality relations and the resulting optimality conditions stated above have a clear economic interpretation (partly given in [8]). These conditions provide a basis for the construction of numerical methods. A straightforward implementation of the optimality conditions leads to iterative methods (see references in [3]). The duality relations are used in finite-step methods, which are considered in the following parts.

References

[1] Dantzig, G.B., *Linear Programming and Extensions*, Princeton University Press, Princeton, N.J., 1963.

[2] Kantorovich, L.V., *The Best Use of Economic Resources*, Harvard University Press, Cambridge, Mass., 1965.

[3] Propoi, A.I., *Problems of Dynamic Linear Programming*, RM-76-78, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1976.

[4] Pontryagin, L.S., et al., *Mathematical Theory of Optimal Processes*, Wiley Interscience, N.Y., 1962.

[5] Propoi, A.I., *Elementi Theorii Optimalnikh Discretnikh Protsessov, (Elements of Theory of Optimal Discrete Processes)*, Nauka, Moscow, 1973 (in Russian).

[6] Invanilov, Yu.P. and A.I. Propoi, On Problems of the Linear Dynamic Programming, *Dokl.Acad.Nauk SSSR*, 198, 5 (1971).

[7] Goldman, A.J. and A.W. Tucker, Theory of Linear Programming, in H.W. Kuhn and A.W. Tucker, eds., *Linear Equalities and Related Systems*, Princeton University Press, No. 38, 1956, 53-98.

[8] Ivanilov, Yu.P. and A.I. Propoi, Duality Relations in Linear Dynamic Programming, *Automation and Remote Control*, No. 12 (1973).

II. THE DYNAMIC SIMPLEX METHOD: GENERAL APPROACH

1. Introduction

Methods for solving general linear programming (LP) are now well developed and have resulted in an extensive field of applications [1,2]. Dynamic linear programming (DLP) is a special class of linear programs for planning and control of complex systems over time [3-6]. DLP applications tend to be too large to be solved by general LP methods. These applications have been hampered by lack of universal DLP computer codes. The few DLP problems that are solved are limited in size. They are solved by treating them as static problems and using for their solution standard LP codes (see, for example, [4,6]).

As DLP problems are principally large-scale, this "static" approach is limited in its possibilities, and the development of efficient algorithms specially oriented to dynamic LP problems continues to be needed. In recent years, methods for DLP have been proposed which make it possible to take into account the specific features of dynamic problems (e.g. [7-9]).* But extension of the most effective finite-step method -- the simplex method for solving LP -- to the dynamic case yet has to be implemented although there have been a number of proposals by Dantzig and others.

The dynamic simplex method as presented here was first suggested in [10,11]. In this approach, the main concept of the static simplex method -- the basis -- is replaced by a set of local bases, introduced for the whole planning period. It allows a significant saving in the amount of computation and computer core. It permits the development of a set of finite-step DLP methods (primal, dual and primal-dual) which are the direct analogues of the corresponding static finite-step methods.

This paper consists of two parts: the first part describes the proposed approach; the second part presents the separate

*See also references in [3].

procedures and the general scheme of the algorithm as well as the connection with the basis factorization approach.

Consider the DLP problem in the following form.

Problem 1.1. Find a control

$$u = \{u(0),\ldots,u(T-1)\}$$

and a trajectory

$$x = \{x(0),\ldots,x(T)\} \quad ,$$

satisfying the state equation

$$x(t+1) = A(t)x(t) + B(t)u(t) \quad (t = 0,1,\ldots,T-1) \tag{1.1}$$

with initial condition

$$x(0) = x^0 \tag{1.2}$$

and constraints

$$G(t)x(t) + D(t)u(t) = f(t) \tag{1.3}$$

$$u(t) \geq 0 \quad \geq \quad (t = 0,1,\ldots,T-1) \tag{1.4}$$

which maximize the objective function

$$J_1(u) = a(T)x(T) \quad . \tag{1.5}$$

Here the vector $x(t) = \{x_1(t),\ldots,x_n(t)\}$ defines the state of the system at step t in the state space E^n, which is assumed to be the n-dimension euclidean space; the vector $u(t) = \{u_1(t), \ldots,u_r(t)\} \in E^r$ (r-dimension euclidean space) specifies the controlling action at step t; vectors x^0, $f(t)$ and the matrices $A(t)$, $B(t)$, $G(t)$, $D(t)$, respectively are of dimensions $(n \times 1)$, $(m \times 1)$ and $(n \times n)$, $(n \times r)$, $(m \times n)$, $(m \times r)$, and are assumed to be given. In vector products, the right vector is a column, the left vector is a row.

There are a number of modifications of Problems 1.1 which can either be reduced to this problem [12,13] or the results stated below may be used directly for their solution. For example, constraints on the state and control variables can be separate; state variables may be nonnegative; state equations include time lags; the objective function depends on the whole sequences $\{u(t)\}$ and/or $\{x(t)\}$, etc. [3,12]).

Along with the primal Problem 1.1, use will be made of its dual [12].

Problem 1.2. Find a dual control

$$\lambda = \{\lambda(T-1),\ldots,\lambda(0)\}$$

and a dual (conjugate) trajectory

$$p = \{p(T),\ldots,p(0)\} \quad ,$$

satisfying the costate (conjugate) equation

$$p(t) = p(t+1)A(t) - \lambda(t)G(t) \quad (t=T-1,\ldots,1,0) \tag{1.6}$$

with boundary condition

$$p(T) = a(T) \tag{1.7}$$

and constraints

$$p(t+1)B(t) - \lambda(t)D(t) \leq 0 \quad (t=T-1,\ldots,1,0) \tag{1.8}$$

which minimize the objective function

$$J_2(\lambda) = p(0)x^0 + \sum_{t=0}^{T-1} \lambda(t)f(t) \quad . \tag{1.9}$$

Definition 1.1. *A feasible control* of the DLP Problem 1.1 is a vector sequence $u = \{u(0),\ldots,u(T-1)\}$ which satisfies with some trajectory $x = \{x(0),\ldots,x(T)\}$ conditions (1.1) to (1.4).

An *optimal control* of Problem 1.1 is a feasible control u^*, which maximizes (1.5) subject to (1.1) - (1.4).

Feasible dual control λ *and optimal dual control* λ^* to the dual Problem 1.2 are defined in a similar way.

Let $U = E^{rT}$; $u = \{u(0),\dots,u(T-1)\} \in U$ be the control space of Problem 1.1. In the control space U Problem 1.1 can be re-written as follows [13].

One can obtain from the state equation (1.1), that

$$x(t) = \Psi(t,0)x(0) + \sum_{\tau=0}^{t-1} \Psi(t,\tau+1)B(\tau)u(\tau) \qquad (1.10)$$

where

$$\Psi(t,\tau) = A(t-1)\cdot A(t-2)\cdots A(\tau) \qquad (0 \leq \tau \leq t-1) \quad ,$$

$$\Psi(t,t) = I \quad .$$

I is the identity.

By substituting (1.10) into (1.3) and taking into account (1.2), we obtain the constraints on controls u, given in explicit form:

$$\sum_{\tau=0}^{t} W(t,\tau)u(\tau) = h(t) \quad , \qquad (1.11)$$

$$u(t) \geq 0 \quad , \quad (t = 0,\dots,T-1) \quad .$$

Here

$$W(t,\tau) = G(t)\Psi(t,\tau+1)B(\tau) \qquad (t > \tau) \quad ,$$

$$W(t,t) = D(t) \quad , \qquad h(t) = f(t) - G(t)\Psi(t,0)x^0 \quad .$$

The matrices $W(t,\tau)$ are of dimension $(m \times r)$ and vectors $h(t)$ are of dimension $(m \times 1)$.

The objective function (1.5) will be rewritten, respectively, in the form

$$J_1(u) = \sum_{t=0}^{T-1} c(t)u(t) + q(0)x^0 \quad , \tag{1.12}$$

where

$$c^T(t) = q(t+1)B(t) \quad .$$

Here vectors $q(t)$ are generated recursively by

$$q(t) = q(t+1)A(t) \qquad (t = T-1,\ldots,0)$$

$$q(T) = a(T) \quad .$$

Denoting the constraint matrix of (1.11) by W (dimension is $mT \times rT$), we can reformulate Problem 1.1 in the following equivalent form.

Problem 1.1a. Find a control u, satisfying the constraints

$$Wu = h \qquad u \geq 0 \quad ,$$

which maximizes the objective function

$$\tilde{J}_1(u) = cu \quad .$$

Here $u = \{u(t)\}$; $h = \{h(t)\}$; $c = \{c(t)\}$ $(t = 0,1,\ldots,T-1)$ and $\tilde{J}_1$ differs from J_1 by the constant $q(0)x^0$.

It is evident that the sets of optimal controls for Problem 1.1 and 1.1a are the same.

Now the general scheme of the simplex method as applied to Problems 1.1a will be described.

Let u be a feasible control; we shall define the index sets

$$I(u) = \{(i,t) \mid u_i(t) > 0 \,;\, i = 1,\ldots,r \,;\, t = 0,\ldots,T-1\}$$

$$\bar{I}(u) = \{(i,t) \mid u_i(t) = 0 \,;\, i = 1,\ldots,r \,;\, t = 0,\ldots,T-1\}$$

$$I = I(u) \cup \bar{I}(u) \quad .$$

Denote also the columns of matrix W by $w_i(t)$ $(i = 1,\dots,r;$ $t = 0,1,\dots,T-1;$ $w_i(t) \in E^{mT}$. In this case the constraints (1.11) can be rewritten as

$$\sum_{(i,t)\in I} w_i(t)u_i(t) = h \quad ; \qquad u_i(t) \geq 0 \quad .$$

<u>*Definition 1.2.*</u> A *basic feasible control* of Problem 1.1 is a feasible control u, for which vectors $w_i(t)$, $(i,t) \in I(u)$, are linearly independent.

A *nondegenerate basic feasible control* is a basic feasible control u, for which vectors $w_i(t)$, $(i,t) \in I(u)$, constitute a basis in E^{mT}.

The *basis of a basic feasible control* u is a system of mT linearly independent vectors $w_i(t)$, which contains all vectors $w_i(t)$, $i(t) \in I(u)$.

As usual without any loss in generality we can assume that Problem 1.1a (1.1) is feasible and that any basic feasible control is nondegenerate [1].

Denote by $I_B(u)$ the set of indices corresponding to the basic vectors $w_i(t)$; $I_N(u)$ is the set of indices corresponding to the remaining vectors $w_i(t)$ of matrix W. Let

$$u_B = \{u_i(t) \,|\, (i,t) \quad I_B(u)\} \quad ,$$

$$u_N = \{u_i(t) \quad (i,t) \quad I_N(u)\} \quad ,$$

and m(t) is the number of basic components of a basic control u at step t. Evidently

$$\sum_{t=0}^{T-1} m(t) = mT \qquad .$$

Then, any basic feasible control may be represented as

$$u = \{u_B, u_N\}, \quad \text{with } u_B \geq 0 \quad , \qquad u_N = 0 \quad .$$

Denote by W_B the matrix with columns $w_i(t)$, $(i,t) \in I_B(u)$ (basic matrix). Then $u_B = W_B^{-1}h$.

Let $w_j(t_1), (j,t_1) \in I$, be an arbitrary column vector of W, then

$$w_j(t_1) = W_B v_j(t_1) \quad , \tag{1.13}$$

where vector $v_j(t_1) = \{v_{ij}(t_1,\tau)\}$, $(i = 1,\ldots,m\ ,\ \tau = 0,\ldots,T-1)$ has dimension mT.

Define

$$z_j(t_1) = c_B v_j(t_1) \quad .$$

Thus, we can rewrite

$$\begin{aligned} c_j(t) &= q(t+1)b_j(t) \\ z_j(t) &= \sum_{\tau=0}^{T-1} q(\tau+1)B_B(\tau)v_j(t,\tau) \quad . \end{aligned} \tag{1.14}$$

Here $b_j(t)$ is a column of the matrix $B(t)$; the matrix $B_B(\tau)$ is generated by the basic columns $b_i(\tau)$, $(i,\tau) \in I_B(u)$ of the matrix $B(\tau)$; $(j,t) \in I$.

The direct application of the simplex method to Problem 1.1 (1.1a) gives the following basic operations:

1. The computation of the sign s or $z_j(t) - c_j(t)$ for all $(j,t) \in I$, to determine whether an optimal control has been found; that is the case when $z_j(t) - c_j(t) \geq 0$ for all j and t. If yes, the algorithm terminates with a printout of the optimal solution. If not, then

2. the selection of the vector to be introduced into the basis, that is selection of a vector with a value of $z_j(t) - c_j(t) < 0$. Let the pair of indices associated with this vector be (j,t_1).

3. The selection of the vector to be removed from the basis. The pair of indices associated with this vector will be denoted by (ℓ, t_2). If (ℓ, t_2) cannot be found, the algorithm terminates with a printout of information of how to generate a class of feasible solutions such that $J_1(u) \to +\infty$. If not, then

4. the basis and basic feasible control is updated. The new basic feasible control $u^{(1)} = \{u_B^{(1)}, 0\}$ is defined by

$$
\begin{aligned}
& u_{s_i}^{(1)}(\tau) = u_{s_i}(\tau) - \theta_0 v_{s_{i_j}}(t_1, \tau) \qquad (s_i, \tau) \in I_B(u) \\
& u_j^{(1)}(t_1) = \theta_0 \qquad\qquad (1.15) \\
& u_i^{(1)}(\tau) = 0 \qquad (i, \tau) \neq (j, t_1); \qquad (i, \tau) \in I_N(u) \quad ,
\end{aligned}
$$

where the outgoing pair of indices (ℓ, t_2) is given by the value θ_0 which is calculated from

$$
(\ell, t_2) = \underset{\substack{v_{s_{i_j}}(t_1, \tau) > 0 \\ (s_i, \tau) \in I(u)}}{\text{arg-min}} \frac{u_{s_i}(\tau)}{v_{s_{i_j}}(t_1, \tau)} \qquad (1.16)
$$

and θ_0 by

$$
\theta_0 = \frac{u_{s_\ell}(t_2)}{v_{\ell j}(t_1, t_2)} \quad .
$$

The numbers $z_j(t)$ are usually computed from $z_j(t) = \lambda w_j(t)$, where $\lambda = \{\lambda_i(\tau), (i, \tau) \in I_B(u)\}$ are simplex multipliers for the basis W_B:

$$
\lambda = c_B W_B^{-1} \quad . \qquad (1.17)
$$

The general scheme considered above is in practice ineffective for the solution of Problem 1.1 (1.1a) when the dimension of the matrix W is large. Besides, the input data are usually given in the form of Problem 1.1 rather than in the form of Problem 1.1a and no exploitation has been made of its special structure. Therefore the simplex procedure directly designed for the solution of Problem 1.1 will be described.

2. Local Bases

The matrices $D(t)$ $(t = 0,\dots,T-1)$ of constraints (1.3) will be assumed to have the rank m. This assumption is not restrictive because one could always insert, if necessary, additional artificial columns, as in the static case, see [1].

Let us denote $\hat{f}(0) = f(0) - G(0)x^0$. Then constraints (1.3) can be rewritten as

$$D(0)u(0) = \hat{f}(0) \quad . \tag{2.1}$$

In accordance with our assumption we can choose m linearly independent column-vectors $d_i(0)$ of the matrix $D(0)$. Denote these columns by $D_0(0)$ and the rest of $D(0)$ by $D_1(0)$. Thus

$$D(0) = [D_0(0);D_1(0)] \quad .$$

As determinant $|D_0(0)| \neq 0$, the constraints (2.1) can be rewritten in the form

$$u_0(0) = D_0^{-1}(0)\hat{f}(0) - D_0^{-1}(0)D_1(0)u_1(0) \quad , \tag{2.2}$$

where components of the vector $u_0(0) \in E^m$ correspond to the matrix $D_0(0)$ and components of the vector $u_1(0) \in E^{r-m}$ correspond to the matrix $D_1(0)$.

The partition of the matrix $B(0)$ is carried out similarly to that of the partition of $D(0)$: $B(0) = [B_0(0);B_1(0)]$. Therefore

$$x(1) = A(0)x(0) + B_0(0)u_0(0) + B_1(0)u_1(0) \quad . \tag{2.3}$$

Substitution (2.2) into (2.3) yields

$$x(1) = x^*(1) + B^1(0)u_1(0) \quad , \tag{2.4}$$

where

$$B^1(0) = B_1(0) - B_0(0)D_0^{-1}(0)D_1(0) \quad ,$$
$$x^*(1) = A(0)x^0 + B_0(0)u_0^*(0) \quad ,$$
$$u_0^*(0) = D_0^{-1}(0)\hat{f}(0) \quad .$$

Now we consider a step t, $0 \leq t \leq T-1$. Let

$$\hat{D}(t)\hat{u}(t) = \hat{f}(t) \tag{2.5}$$

where

$$\hat{D}(t) = [G(t)B^1(t-1);D(t)] \tag{2.6}$$
$$\hat{u}(t) = [\hat{u}_1(t-1);u(t)]^T \tag{2.7}$$
$$\hat{f}(t) = f(t) - G(t)x^*(t) \quad . \tag{2.8}$$

In (2.6) to (2.8), the matrix $B^1(t-1)$ and vectors $\hat{u}_1(t-1)$, $x^*(t)$ are defined from recurrent relations, which will be obtained below.

By construction, the matrix $\hat{D}(t)$ contains m linearly independent columns $\hat{d}_i(t)$. The matrix formed by these columns will be denoted as $\hat{D}_0(t)$; the matrix from the rest of the columns -- as $\hat{D}_1(t)$. Thus, (2.5) can be rewritten as

$$\hat{D}_0(t)\hat{u}_0(t) + \hat{D}_1(t)\hat{u}_1(t) = \hat{f}(t)$$
$$\hat{D}(t) = [\hat{D}_0(t);\hat{D}_1(t)] \quad .$$

Hence

$$\hat{u}_0(t) = \hat{D}_0^{-1}(t)\hat{f}(t) - \hat{D}_0^{-1}(t)\hat{D}_1(t)\hat{u}_1(t) \quad . \tag{2.9}$$

Or

$$\hat{u}_0(t) = \hat{u}_0^*(t) - \Phi(t)\hat{u}_1(t) \quad , \tag{2.10}$$

where

$$\hat{u}_0^*(t) = \hat{D}_0^{-1}(t)\hat{f}(t) \quad , \tag{2.11}$$

$$\Phi(t) = \hat{D}_0^{-1}(t)\hat{D}_1(t) \quad . \tag{2.12}$$

Let

$$x(t) = x^*(t) + B^1(t-1)\hat{u}_1(t-1) \quad , \tag{2.13}$$

where $x^*(t)$ and $B^1(t-1)$ will be defined later.

By substituting (2.14) into state equation (1.1), we obtain

$$x(t+1) = A(t)x^*(t) + \hat{B}(t)\hat{u}(t) \quad , \tag{2.14}$$

where

$$\hat{B}(t) = [A(t)B^1(t-1);B(t)] \quad , \tag{2.15}$$

the vector $\hat{u}(t)$ is defined by (2.7).

Considering the representation

$$\hat{B}(t) = [\hat{B}_0(t);\hat{B}_1(t)] \quad ,$$
$$\hat{u}(t) = [\hat{u}_0(t);\hat{u}_1(t)]^T$$

and substituting (2.10) into (2.14), we again obtain equations (2.13) for the next step $t+1$:

$$x(t+1) = x^*(t+1) + B^1(t)\hat{u}_1(t) \quad ,$$

where

$$x^*(t+1) = A(t)x^*(t) + \hat{B}_0(t)\hat{u}_0^*(t) \quad , \tag{2.16}$$

$$B^1(t) = \hat{B}_1(t) - \hat{B}_0(t)\Phi(t) \quad . \tag{2.17}$$

Initial conditions for (2.14), (2.5) are

$$x^*(0) = x(0); \quad \hat{u}(0) = u(0) \quad ,$$
$$\hat{B}(0) = B(0) \quad , \quad \hat{D}(0) = D(0) \quad . \tag{2.18}$$

The specific of such a representation of Problem 1.1 is a recurrent determination of control $\hat{u}(t)$, that is, using (2.7) we obtain

$$\begin{aligned} \hat{u}(t) &= [\hat{u}_1(t-1), u(t)]^T \qquad\qquad (2.19) \\ &= [\hat{u}_2(t-2), u_1(t-1), u(t)]^T = \ldots = [u_t(0), u_{t-1}(1), \ldots, u_{t-i}(i), \\ &\quad \ldots, u_1(t-1), u(t)]^T \end{aligned}$$

where the vector $u_{t-i}(i)$ is formed from those components of the control u which are recomputed from a step i to the step t by virtue of the procedure which was described above. The relations (2.19) show that the vector $\hat{u}(t)$ may include components $u_i(\tau)$ from preceding steps $\tau = t-1, \ldots, 1, 0$.

Consider now the last step

$$\hat{D}_0(T-1)\hat{u}_0(T-1) + \hat{D}_1(T-1)\hat{u}_1(T-1) = \hat{f}(T-1)$$

where $\hat{D}_0(T-1)$ is a nonsingular matrix. Let

$$\hat{u}_1(T-1) = 0 \quad , \tag{2.20}$$

then

$$\hat{u}_0(T-1) = \hat{D}_0^{-1}(T-1)\hat{f}(T-1) \quad . \tag{2.21}$$

Determining the value of the vector $\hat{u}(T-1) = [\hat{u}_0(T-1), \hat{u}_1(T-1)]^T$ from (2.20), (2.21), one can compute the values of feasible control $\{u(t)\}$ for a given set of local bases $\{\hat{D}_0(t)\}(t = 0, 1, \ldots, T-1)$. This procedure will be called Procedure 1.

Definition 2.1: The set of m linearly independent columns $\hat{d}_i(t)$ of the matrix $\hat{D}(t)$ is called the *local basis* at the step t $(t = 0, 1, \ldots, T-1)$.

The set of all indices (i,t) associated with the components of local basis matrix $\hat{D}_0(t)$ $(t = 0,\ldots,T-1)$ will be denoted by $I_0(u)$, and its complement with respect to I will be denoted by $\bar{I}_0(u)$.

Theorem 2.1: Let a control u *be computed from Procedure 1 for a given set of local bases* $\{\hat{D}_0(t)\}$ *with boundary conditions*

$$\hat{u}_0(T-1) = \hat{D}^{-1}(T-1)\hat{f}(T-1)$$
$$\hat{u}_1(T-1) = 0$$

and let

$$u_i(t) \geq 0 \qquad \text{for all} \qquad (i,t) \in I_0(u) \quad .$$

Then u *is a basic feasible control and*

$$u = \{u_B, u_N\} \quad ,$$
$$u_B = \{u_i(t) \mid (i,t) \in I_0(u)\} \quad ,$$
$$u_N = \{u_i(t) \mid (i,t) \in \bar{I}_0(u)\} \quad .$$

Proof: Let W_0 be the matrix which is generated by the columns $w_i(t)$ of the constraint matrix W, associated with variables $\hat{u}_0(t)$, that is,

$$W_0 = \|w_i(t)\| \quad , \qquad (i,t) \in I_0(u) \quad .$$

By construction, W_0 is a square matrix of dimension of $mT \times mT$.

For proof of the theorem, we shall need the following assertion.

Lemma 2.1: The matrix W_0 *is nonsingular if and only if the matrices* $\hat{D}_0(t)$ $(t = 0,1,\ldots,T-1)$ *are nonsingular.*

Proof: *Sufficiency.* The procedure of computing $\{\hat{u}_0(t)\}$ described above is a block modification of the Gauss method [14] where pivot blocks are matrices $\hat{D}_0(t)$. The Gauss algorithm transforms the matrix W_0 to an upper block triangular matrix with $\hat{D}_0(t)$ on its diagonal:

$$W_0 = \begin{bmatrix} \hat{D}_0(0) & * & & & & \\ & \hat{D}_0(1) & * & & 0 & \\ & & \ddots & & & \\ & & & \hat{D}_0(t) & * & \\ & 0 & & & \ddots & \\ & & & & & \hat{D}_0(T-1) \end{bmatrix}$$

where nonzero elements of W_0 are denoted by $*$.

The Gauss algorithm does not change the rank of the original matrix [14]. In fact, the relation

$$\left|\,|W_0|\,\right| = \left|\,|D_0(0)| \ \ldots \ |D_0(T-1)|\,\right| \tag{2.22}$$

holds, where $\left|\,|W_0|\,\right|$ is the absolute value of the determinant of a matrix W_0. The relation (2.22) implies that, if matrices $\hat{D}_0(t)$ $(t = 0,1,\ldots,T-1)$ are nonsingular, then the matrix W_0 is also nonsingular.

Necessity: Suppose that k iterations of the Gauss algorithm have been done and W_0^k is a matrix obtained after k iterations:

$$W_0^k = \begin{bmatrix} \hat{D}_0(0) & * & & & \\ & \hat{D}_0(1) & * & 0 & \\ & & \ddots & & \\ & 0 & & \hat{D}_0(k-1) & * \\ & & & & \ddots \\ & & & & & \tilde{W}_0^k \end{bmatrix}$$

Here $\tilde{W}_0^k$ is a submatrix, generated by elements of W_0^k which are outside of pivot rows and columns of previous iterations. In this case, the relation (2.22) should be replaced by

$$\left| |W_0| \right| = \left| |\hat{D}_0(0)| \ \ldots \ |\hat{D}_0(k-1)| \ |\tilde{W}_0^k| \right| \quad .$$

The first block-row of $\tilde{W}_0^k$ is $[\hat{D}(k);0]$. Suppose that the matrix $\hat{D}(k)$ cannot generate any nonsingular square submatrix $\hat{D}_0(k)$ of dimension m. This implies that the rows of the matrix $\hat{D}(k)$ are linearly dependent and the matrix $\tilde{W}_0^k$ is singular with $|\tilde{W}_0^k| = 0$. Then $|W_0| = 0$, which contradicts the assumption of the lemma.

Thus, if the matrix W_0 is nonsingular then at each step of the Gauss algorithm a nonsingular matrix $\hat{D}_0(k)$ can be constructed. This completes the proof of the lemma.

The proof of the theorem now follows directly. By definition, matrices $\hat{D}_0(t)\,(t = 0,\ldots,T-1)$ are nonsingular, which implies that the matrix W_0 is also nonsingular and vectors $w_i(t)$, $(i,t) \in I_0(u)$, are linearly independent.

It follows from Procedure 1 that

$$u_i(t) = 0 \quad \text{for all} \quad (i,t) \in \bar{I}_0(u) \quad .$$

As $u_i(t) \geq 0$ for all $(i,t) \in I_0(u)$, then in accordance with definition 1.2 u is a basic feasible control. This completes the proof of the theorem.

The proof of Theorem 3.1 shows that Procedure 1 permits operations not with the inverse W_B^{-1} of dimension $mT \times mT$ but with T inverses $\hat{D}_0^{-1}(t)$ of dimension $m \times m$. Hence, Procedure 1 is basic to this approach. However, as will be seen further, it is not used explicitly.

In fact, as follows from the proof of the theorem, only basic submatrices of matrices $\hat{D}(t)$ should be handled in the algorithm. Besides, there is no necessity to compute local bases at each iteration, only the updating of some of the T local bases is needed.

3. Control Variation

In accordance with Theorem 2.1, the basis W_B is equivalent to the set of local bases $\{\hat{D}_{0B}(t)\}$. Therefore, our aim is to develop the simplex operations for solution of Problem 1.1 relative to the set of local bases $\{\hat{D}_{0B}(t)\}$.

For a given basic feasible control $u = \{u_B, u_N\}$, let us fix the pair of indices $(j,t_1) \in I$ such that the corresponding column $d_j(t_1)$ of the matrix $D(t_1)$ is not in the basis, that is, $(j,t_1) \in I_N(u)$.

We first consider the procedure for selection of the column $d_j(t_1)$ to be introduced into the basis, that is, into the set of local bases $\{\hat{D}_{0B}(t)\}$. In accordance with Section 2, the constraints (1.3) at step t can be written as

$$\hat{D}_{0B}(t)\hat{u}_{0B}(t) + \hat{D}_{1B}(t)\hat{u}_{1B}(t) = \hat{f}(t) \tag{3.1}$$

where

$$[\hat{D}_{0B}(t);\hat{D}_{1B}(t)] = \hat{D}_B(t) \quad ,$$
$$[\hat{u}_{0B}(t);\hat{u}_{1B}(t)] = \hat{u}_B(t) \quad , \qquad \hat{u}_B(t) \geq 0 \quad .$$

Here the subscript B denotes submatrices and vectors associated with a given basis W_B or $\{\hat{D}_{0B}(t)\}$.

Let a vector $\hat{v}^*_{0B}(t_1) \in E^m$ define representation of the vector $d_j(t_1)$ in terms of column-vectors of the matrix $\hat{D}_{0B}(t_1)$, that is,

$$\hat{v}^*_{0B}(t_1) = \hat{D}^{-1}_{0B}(t_1)d_j(t_1) \quad . \tag{3.2}$$

Taking into account (3.2), we can rewrite (3.1) as

$$\hat{D}_{0B}(t_1)\left[\hat{u}_{0B}(t_1) - \theta\hat{v}^*_{0B}(t_1)\right] + \hat{D}_{1B}(t_1)\hat{u}_{1B}(t_1) + \theta d_j(t_1) = \hat{f}(t_1) \tag{3.3}$$

where θ is a real number.

It is evident that the equality (3.3) is true for any value of the parameter θ. It follows from (3.3) that a new control $u^{\theta}(t_1)$ is introduced at step t_1:

$$\hat{u}^{\theta}(t_1) = \left[\hat{u}^{\theta}_{0B}(t_1);\hat{u}^{\theta}_{1B}(t_1);\hat{u}^{\theta}_{N}(t_1)\right]^T \quad ,$$

where

$$\begin{aligned} \hat{u}^{\theta}_{0B}(t_1) &= \hat{u}_{0B}(t_1) - \theta\hat{v}^{*}_{0B}(t_1) \\ u^{\theta}_{1B}(t_1) &= \hat{u}_{1B}(t_1) \\ \hat{u}^{\theta}_{N}(t_1) &= [0,\ldots,\theta,\ldots,0]^T \quad . \end{aligned} \tag{3.4}$$

By substituting the control $\hat{u}^{\theta}(t_1)$ in state equation (2.14), we obtain

$$x^{\theta}(t_1 + 1) = x(t_1 + 1) - \theta y^{*}(t_1 + 1) \quad , \tag{3.5}$$

where

$$\begin{aligned} x(t_1 + 1) &= x^{*}(t_1 + 1) + B^{1}_{B}(t_1)\hat{u}_{1B}(t_1) \quad , \\ y^{*}(t_1 + 1) &= \hat{B}_{0B}(t_1)\hat{v}^{*}_{0B}(t_1) - b_j(t_1) \quad . \end{aligned} \tag{3.6}$$

Substituting (3.5) into formulation (2.5) of constraints (1.3), we see that they will be true if

$$\hat{D}_B(t_1 + 1)\hat{u}^{\theta}_B(t_1 + 1) - \theta G(t_1 + 1)y^{*}(t_1 + 1) = \hat{f}(t_1 + 1) \quad . \tag{3.7}$$

Let us express the vector $-G(t_1 + 1)y^{*}(t_1 + 1)$ in terms of column vectors of the matrix $\hat{D}_{0B}(t_1 + 1)$:

$$\hat{v}^{*}_{0B}(t_1 + 1) = -\hat{D}^{-1}_{0B}(t_1 + 1)G(t_1 + 1)y^{*}(t_1 + 1) \quad . \tag{3.8}$$

Considering (3.8), the equality (3.7) can be rewritten as

$$\begin{aligned} \hat{D}_{0B}(t_1 + 1) &\left[\hat{u}_{0B}(t_1 + 1) - \theta\hat{v}^{*}_{0B}(t_1 + 1)\right] \\ &+ \hat{D}_{1B}(t_1 + 1)\hat{u}_{1B}(t_1 + 1) - \theta G(t_1 + 1)y^{*}(t_1 + 1) \\ &= \hat{f}(t_1 + 1) \quad . \end{aligned}$$

We see that the introduction of the compensating term into the equality (3.7) is equivalent to the introduction of a new control $\hat{u}^{\theta}(t_1+1)$ at step t_1+1:

$$\hat{u}^{\theta}(t_1+1) = [\hat{u}^{\theta}_{0B}(t_1+1);\hat{u}^{\theta}_{1B}(t_1+1);\hat{u}^{\theta}_{N}(t_1+1)] \quad ,$$

where

$$\begin{aligned}\hat{u}^{\theta}_{0B}(t_1+1) &= \hat{u}_{0B}(t_1+1) - \theta\hat{v}^{*}_{0B}(t_1+1)\\ \hat{u}^{\theta}_{1B}(t_1+1) &= \hat{u}_{1B}(t_1+1)\\ \hat{u}^{\theta}_{N}(t_1+1) &= 0 \quad .\end{aligned} \tag{3.9}$$

Thus, the variation of the control (3.4) at step t_1, where vector $\hat{v}^{*}_{0B}(t_1)$ is defined by (3.2), induces a variation of control (3.9) at the next steps $\tau = t_1+1,\ t_1+2,\dots,T-2$ with

$$\hat{v}^{*}_{0B}(\tau) = -\hat{D}^{-1}_{0B}(\tau)G(\tau)y^{*}(\tau) \quad . \tag{3.10}$$

Vectors $y^{*}(\tau)$ are satisfied to the following difference equation:

$$y^{*}(\tau+1) = A(\tau)y^{*}(\tau) + \hat{B}_{0B}(\tau)\hat{v}^{*}_{0B}(\tau) \tag{3.11}$$

where vectors $\hat{v}^{*}_{0B}(\tau)$ $(\tau = t_1+1,\dots,T-1)$ are defined from (3.10) and vector $\hat{v}^{*}_{0B}(t_1)$ is defined from (3.2).

Now we consider the last step:

$$\begin{aligned}\hat{D}_{B}(T-1)\left[\hat{u}_{B}(T-1) - \theta\hat{v}_{B}(T-1)\right] - \theta G(T-1)y^{*}(T-1)\\ = \hat{f}(T-1) \quad .\end{aligned} \tag{3.12}$$

As $u = \{u_B, 0\}$ is a basic feasible control, then by virtue of Theorem 2.1, the matrix $\hat{D}_{B}(T-1)$ is nonsingular and

$$\hat{D}_{B}(T-1) = \hat{D}_{0B}(T-1) \quad .$$

Therefore (3.12) yields that

$$\hat{v}_B(T-1) = \hat{v}^*_{0B}(T-1) = -\hat{D}^{-1}_{0B}(T-1)G(T-1)\underline{y}^*(T-1) \quad .$$

By construction, the structure of vector $\hat{v}_B(T-1)$ is similar to the structure of vector $\hat{u}_B(T-1)$. Hence, define a vector:

$$\hat{v}_B(T-1) = \left[\hat{v}_{1B}(T-2), v_B(T-1)\right] \tag{3.13}$$

where vector $v_B(T-1)$ is associated with the variation of vector $u_B(T-1)$, vector $\hat{v}_{1B}(T-2)$ is associated with the variation of vector $\hat{u}_{1B}(T-2)$:

$$\hat{u}^{\theta}_{1B}(T-2) = \hat{u}_{1B}(T-2) - \theta\hat{v}_{1B}(T-2) \quad .$$

To satisfy the constraints at step $T-2$, the additional term $-\theta\hat{D}_{1B}(T-2)v_{1B}(T-2)$ must be compensated by the additional variation $\hat{v}^1_{0B}(T-2)$ of control $\hat{u}_{0B}(T-2)$:

$$\hat{u}^{\theta}_{0B}(T-2) = \hat{u}_{0B}(T-2) - \theta\left[\hat{v}^*_{0B}(T-2) - \hat{v}^1_{0B}(T-2)\right] \quad ,$$

where

$$\hat{v}^1_{0B}(T-2) = \hat{D}^{-1}_{0B}(T-2)\hat{D}_{1B}(T-2)\hat{v}_{1B}(T-2) = \Phi_B(T-2)\hat{v}_{1B}(T-2) \quad .$$

Let $\hat{v}_{0B}(T-2) = \hat{v}^*_{0B}(T-2) - \hat{v}^1_{0B}(T-2)$. As in the case of (2.7), we can write

$$\begin{aligned} \hat{v}_B(T-2) &= \left[\hat{v}_{0B}(T-2), \hat{v}_{1B}(T-2)\right] \\ &= \left[\hat{v}_{1B}(T-3), v_B(T-2)\right] \quad . \end{aligned} \tag{3.14}$$

By induction, we find that in order to satisfy the constraints (1.2) for all θ and $\tau = 0,1,\ldots,T-1$, we must define

$$\hat{D}_B(T-1)[\hat{u}_B(T-1) - \theta\hat{v}_B(T-1)] - \theta G(T-1)y^*(T-1) = \hat{f}(T-1)$$

$$\hat{D}_{0B}(\tau)[\hat{u}_{0B}(\tau) - \theta(\hat{v}^*_{0B}(\tau) - \hat{v}^1_{0B}(\tau))] + \hat{D}_{1B}(\tau)[\hat{u}_{1B}(\tau) - \theta\hat{v}_{1B}(\tau)]$$
$$- \theta G(\tau)y^*(\tau) = \hat{f}(\tau) \quad \text{if} \quad t_1 + 1 \le \tau \le T - 2 \quad ,$$

$$\hat{D}_{0B}(t_1)[\hat{u}_{0B}(t_1) - \theta(\hat{v}^*_{0B}(t_1) - \hat{v}^1_{0B}(t_1))]$$
$$+ \hat{D}_{1B}(t_1)[\hat{u}_{1B}(t_1) - \theta\hat{v}_{1B}(t_1)] + \theta d_j(t_1) = \hat{f}(t_1) \tag{3.15}$$

$$\hat{D}_{0B}(\tau)[\hat{u}_{0B}(\tau) + \theta\hat{v}^1_{0B}(\tau)] + \hat{D}_{1B}(\tau)[\hat{u}_{1B}(\tau) - \theta\hat{v}_{1B}(\tau)]$$
$$= \hat{f}(\tau) \quad \text{if} \quad 0 \le \tau \le t_1 - 1 \quad .$$

The vectors $\hat{v}^*_{0B}(\tau)$must satisfy the following relations:

$$\hat{v}^*_{0B}(T-1) = -\hat{D}^{-1}_{0B}(T-1)G(T-1)y^*(T-1) = \hat{v}_B(T-1) \quad ,$$

$$\hat{v}^*_{0B}(\tau) = -\hat{D}^{-1}_{0B}(\tau)G(\tau)y^*(\tau) \quad \text{if} \quad t_1 + 1 \le \tau \le T-2 \quad ,$$

$$\hat{v}^*_{0B}(t_1) = \hat{D}^{-1}_{0B}(t_1)d_j(t_1) \quad .$$

The vectors $\hat{v}^1_{0B}(\tau)$ satisfy the relations $(0 \le \tau \le T-2)$:

$$\hat{v}^1_{0B}(\tau) = \hat{D}^{-1}_{0B}(\tau)\hat{D}_{1B}(\tau)\hat{v}_{1B}(\tau) = \Phi_B(\tau)\hat{v}_{1B}(\tau) \quad .$$

Thus the variation $\hat{v}_{0B}(\tau)$ of control $\hat{u}_{0B}(\tau)$ $(\tau = 0,1,\ldots,T-1)$ is defined by:

$$\begin{aligned} \hat{v}_{0B}(T-1) &= \hat{v}^*_{0B}(T-1) \quad , \\ \hat{v}_{0B}(\tau) &= \hat{v}^*_{0B}(\tau) - \hat{v}^1_{0B}(\tau) \, , \ \text{if} \ t_1 + 1 \le \tau \le T-2 \\ \hat{v}_{0B}(\tau) &= -\hat{v}^1_{0B}(\tau) \quad , \quad \text{if} \ 0 \le \tau \le t_1 \quad . \end{aligned} \tag{3.16}$$

Using (3.12) and (3.13) we can define the values of vectors $\{v_B(\tau)\}$ associated with the variation of control $\{u_B(\tau)\}$. Thus, if a new column $w_j(t_1)$ associated with a column $d_j(t_1)$ is introduced into the basis W_B, then the variation of a basic feasible control $\{u_B, u_N\}$ is defined by (cf. (1.15)):

$$\hat{u}^{\theta}_{0B}(\tau) = \hat{u}_{0B}(\tau) - \theta\hat{v}_{0B}(\tau) \quad . \tag{3.17}$$

We shall refer to the determining of the variation $\{\hat{u}^{\theta}(\tau)\}$ of a feasible control $\{\hat{u}(t)\}$ as Procedure 2. The variation $\{\hat{u}^{\theta}(\tau)\}$ is satisfied to the constraints (1.1) to (1.3) of Problem 1.1 by definition. As $\{\hat{u}(\tau)\}$ is a feasible control, then the constraints (1.4) will also be satisfied for sufficiently small $\theta \geq 0$. Hence the control $\{\hat{u}^{\theta}(\tau)\}$ is feasible if $0 \leq \theta \leq \theta_0$. The value of θ_0 is defined by relations (cf. (1.16)):

$$(\ell, t_2) = \text{arg-min} \frac{\hat{u}_{0i}(\tau)}{\hat{v}_{0i}(\tau)} \quad ; \tag{3.18}$$

$$\theta_0 = \frac{\hat{u}_{0\ell}(t_2)}{\hat{v}_{0\ell}(t_2)} \quad ,$$

where the minimum if taken over all $(i,\tau) \in I_0(u), \hat{v}_{0i}(\tau) > 0$ and $\hat{u}_{0i}(\tau)$, $\hat{v}_{0i}(\tau)$ are the i-th components of vectors $\hat{u}_{0B}(\tau)$, $\hat{v}_{0B}(\tau)$.

The equality (3.18) follows from (1.4) and (3.16); minimum in (3.18) is achieved at single pair (ℓ, t_2) in the nondegenerate case.

Let us now define the variation of trajectory $\{x(t)\}$. Considering (3.5), (3.13) and (3.15), we find that the variation of trajectory $x^{\theta}(\tau) = x(\tau) - \theta y(\tau)$ $(\tau = 1,\dots,T)$ will be defined by

$$\begin{aligned} y(T) &= y^{*}(T) \\ y(\tau+1) &= y^{*}(\tau+1) + B^{1}_{B}(\tau)\hat{v}_{1B}(\tau) \qquad (\tau = T-2,\dots,1,0) \end{aligned} \tag{3.19}$$

where the vectors $y^{*}(\tau) = 0$ if $0 \leq \tau \leq t_1$, and $y^{*}(\tau+1) = A(\tau)y^{*}(\tau) + \hat{B}_{0B}(\tau)\hat{v}^{*}_{0B}(\tau)$, if $t_1 + 1 \leq \tau \leq T-1$.

4. Objective Function Variation

The special feasible variation of a basic feasible control has been built up in the previous section. Now we determine the corresponding variation of the objective function (1.5) when a column vector $d_j(t_1)$, $(j,t_1) \in I_N(u)$ is introduced in the basis W_B.

In accordance with (3.19),

$$J_1(u^\theta) = a(T)x(T) - \theta a(T)y^*(T) \quad .$$

Denote the variation of the objective function by

$$\Delta_j(t_1) \equiv \Delta J_1(u^\theta) = (J_1(u^\theta) - J_1(u))/\theta = a(T)y^*(T) \quad , \quad (4.1)$$

where indices (j,t_1) show that the variation has been caused by introduction of the column $d_j(t_1)$, $(j,t_1) \in I_N(u)$ to the basis.

By substituting $y^*(T)$ from (3.11) with $\tau = T-1$ into (4.1), we obtain

$$\Delta_j(t_1) = a(T)A(T-1)y^*(T-1) + a(T)\hat{B}_{0B}(T-1)\hat{v}^*_{0B}(T-1) \quad . \quad (4.2)$$

Considering (3.16), (2.15) and (1.12), we rewrite (4.2) as

$$\begin{aligned}\Delta_j(t_1) &= q(T-1)y^*(T-1) + q(T-1)B^1_B(T-2)\hat{v}_{1B}(T-2) \\ &\quad + q(T)B_B(T-1)v_B(T-1) \quad ,\end{aligned} \quad (4.3)$$

where $B_B(T-1)$ is the matrix generated by basis columns of the matrix $B(T-1)$, variation $v_B(T-1)$ is associated with basic components of the vector $u_B(T-1)$.

By substituting

$$y^*(T-1) = A(T-2)y^*(T-2) + \hat{B}_{0B}(T-2)\hat{v}^*_{0B}(T-2)$$

into (4.3) and again using (1.12), we obtain

$$\begin{aligned}\Delta_j(t_1) &= q(T-2)y^*(T-2) + q(T-1)\hat{B}_{0B}(T-2)\hat{v}^*_{0B}(T-2) \\ &\quad + q(T-1)B^1_B(T-2)\hat{v}_{1B}(T-2) + q(T)B_B(T-1)v_B(T-1) \quad .\end{aligned}$$

Considering (2.17) and (3.16), we can express $\Delta_j(t_1)$ in the form

$$\Delta_j(t_1) = q(T-2)y^*(T-2) + q(T-1)\hat{B}_{0B}(T-2)\hat{v}_{0B}(T-2)$$
$$+ q(T-1)\hat{B}_{1B}(T-2)\hat{v}_{1B}(T-2) + q(T)B_B(T-1)v_B(T-1) \quad .$$

Hence and from (2.15) it follows that

$$\Delta_j(t_1) = q(T-2)y^*(T-2) + q(T-1)\hat{B}_B(T-2)\hat{v}_B(T-2)$$
$$+ q(T)B_B(T-1)v_B(T-1) \quad .$$

Eventually by induction we obtain for all $(j,t_1) \in I_N(u)$:

$$\Delta_j(t_1) = \sum_{\tau=0}^{T-1} q(\tau+1)B_B(\tau)v_B(\tau) - q(t_1+1)b_j(t_1) \quad . \qquad (4.5)$$

One can see that vectors $v_B(\tau)$ $(\tau = 0,1,\ldots,T-1)$ are a solution of the equations system (1.13). The solution is obtained by means of the compact inverse matrix Procedure 2, which is analogous to Procedure 1 of basic feasible control computation.

Comparing (4.5) and (1.14), we can write

$$\Delta_j(t_1) = z_j(t_1) - c_j(t_1) = \sum_{\tau=0}^{T-1} q(\tau+1)B_B(\tau)v_B(\tau)$$
$$- q(t_1+1)b_j(t_1) \quad .$$

Using the dual Problem 1.2, we can now obtain another form for the definition of the objective function variation $\Delta_j(t_1)$. This form corresponds to (1.17) and is more convenient in practice.

By substituting the expression $\hat{v}^*_{0B}(T-1)$ from (3.10) at $\tau = T-1$ into (4.2), one can obtain

$$\Delta_j(t_1) = a(T)A(T-1)y^*(T-1) - a(T)\hat{B}_{0B}(T-1)\hat{D}^{-1}_{0B}(T-1)G(T-1)y^*(T-1) \quad .$$

Define a vector $\lambda(T-1)$ as $\lambda(T-1) = a(T)\hat{B}_{0B}(T-1)\hat{D}^{-1}_{0B}(T-1)$. Then $\Delta_j(t_1) = p(T-1)y^*(T-1)$, where the vector $p(T-1)$ is computed from dual state equation (1.6) with boundary condition (1.7) at $t = T-1$.

By induction we obtain

$$\Delta_j(t_1) = \lambda(t_1)d_j(t_1) - p(t_1+1)b_j(t_1) \quad , \quad (j,t_1) \in I_N(u) \quad ,$$

where

$$\lambda(t) = p(t+1)\hat{B}_{0B}(t)\hat{D}_{0B}^{-1}(t) \tag{4.6}$$

and the variables $\lambda(t)$, $p(t+1)$ satisfy the dual state equation (1.6) with boundary condition (1.7).

<u>*Theorem 5.1*</u>*: Vectors* $\{\lambda(t)\}$ *computed from (4.6), (1.6) and (1.7) are the simplex-multipliers for the basis* W_B.

Proof: It is sufficient to show, in accordance with the definition of simplex-multipliers [1], that vectors $\lambda(t)$ satisfy the dual constraints (1.8) as equalities for basic indices; that is,

$$p(t+1)b_j(t) - \lambda(t)d_j(t) = 0 \quad , \quad (j,t) \in I_B(u) \quad .$$

For this, let us consider the constraints (1.8) of the dual Problem 2.1 relative to the current basis W_B of the primal Problem 1.1. They can be written at $t=0$ as

$$\lambda(0)D_B(0) = p(1)B_B(0) \quad . \tag{4.7}$$

As a nonsingular matrix $\hat{D}_{0B}(0)$ can be generated by columns of the matrix $D_B(0)$, then (4.7) can be rewritten as

$$\lambda(0)\hat{D}_{0B}(0) = p(1)\hat{B}_{0B}(0)$$
$$\lambda(0)\hat{D}_{1B}(0) = p(1)\hat{B}_{1B}(0) \quad .$$

Now we obtain

$$p(1)\left[\hat{B}_{0B}(0)\hat{D}_{0B}^{-1}(0)\hat{D}_{1B}(0) - \hat{B}_{0B}(0)\right] = 0$$

or, in accordance with (2.17),

$$p(1)B_B^1(0) = 0 \quad . \tag{4.8}$$

Using the state equations (1.6), the conditions (4.8) can be rewritten as

$$p(2)A(1)B_B^1(0) - \lambda(1)G(1)B_B^1(0) = 0 \quad .$$

Hence and from (1.8), we obtain for the next step,

$$\lambda(1) = p(2)\hat{B}_{0B}(1)\hat{D}_{0B}^{-1}(0) \quad .$$

By induction,

$$\lambda(t) = p(t+1)\hat{B}_{0B}(t)\hat{D}_{0B}^{-1}(t)$$

holds for all $t = 1,2,\ldots,T-1$, where matrices $\hat{B}_{0B}(t)$ and $\hat{D}_{0B}^{-1}(t)$ are defined in Section 2. This completes the proof.

Define Procedure 3 by formulas (4.6), (1.6), and (1.7). Procedure 3 allows computation of the values of simplex-multipliers $\{\lambda(t)\}$ for the current basis W_B.

It should be noted that for computing both the values of vectors $\{\lambda(t), p(t+1)\}$ and the values of vectors $\{u(t), x(t)\}$, one can use the same matrices $\hat{D}_{0B}^{-1}(t)$, $\hat{D}_{1B}(t)$, $\hat{B}_{0B}(t)$, and $B_B^1(t)$.

5. Conclusion

As has been shown above, the basis W_B of dimension $mT \times mT$ of the equivalent Problem 1.1a can be replaced by the system of T local bases $\{\hat{D}_{0B}(t)\}$ of dimensions $m \times m$. In this case, all simplex operations (primal, dual solutions, pricing, etc.) can be effectively implemented using this system of local bases.

On the other hand, the original Problem 1.1 can be considered as a structured linear programming problem with constraints (1.1) to (1.4). The basic matrix $\bar{B}$ for this problem has dimension

$(m+n)T \times (m+n)T$. One can easily see that the basic control $u = \{u_B, u_N\}$, determined from Procedure 1 of Section 2 with the corresponding trajectory x, is a basic solution for linear programming Problem 1.1.

The separate operations and the whole algorithm of the dynamic simplex method will be considered in the next part.

References

[1] Dantzig, G.B., *Linear Programming and Extensions*, University Press, Princeton, N.J., 1963.

[2] Kantorovich, L.V., *The Best Use of Economic Resources*, Harvard University Press, Cambridge, Mass., 1965.

[3] Propoi, A.I., *Problems of Dynamic Linear Programming*, RM-76-78, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1976.

[4] Haefele, W. and A.S. Manne, *Strategies of a Transition from Fossil to Nuclear Fuels*, RR-74-07, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1974.

[5] Aganbegian, A.G. and K.K. Valtukh, Izpol'zovanie Narodnokhoziastvennykh Modelei v Planirovanii (Utilization of National Economy Models in Planning), *Ekonomica*, M., (1975) (in Russian).

[6] Swart, W. a.o., Expansion Planning for a Large Dairy Farm, in H. Salkin and J. Saha, eds., *Studies in Linear Programming*, North-Holland, New York, 1975.

[7] Glassey, C.R., Dynamic Linear Programming for Production Scheduling, *Operations Research*, 18, 1 (1970).

[8] Ho, J.K. and A.S. Manne, Nested Decomposition for Dynamic Models, *Mathematical Programming*, 6, 2 (1974).

[9] Propoi, A.I. and A.B. Yadykin, Parametric Iterative Methods for Dynamic Linear Programming. I. Non-Degenerative Case, *Avtomatika i Telemekhanika*, 12, (1975); II. General Case, *Avtomatika i Telemekhanika*, 1, (1976) (in Russian).

[10] Krivonozhko, V.E. and A.I. Propoi, A Method for DLP with the Use of Basis Matrix Factorization, IX International Symposium on Mathematical Programming, Budapest, 1976.

[11] Krivonozhko, V.E. and A.I. Propoi, Method of Successive Control Improvement for DLP, I, II, *Izv.Akad.Nauk SSSR, Technicheskaia Kivernetika*, N3 and N4, 1978(in Russian).

[12] Propoi, A., *Dual Systems of Dynamic Linear Programming*, Part I of this issue.

[13] Propoi, A., *Elementy Teorii Optimalnykh Diskretnykh Protsessov* (Elements of the Theory of Optimal Discrete Processes), Nauka, Moscow, 1973, (in Russian).

[14] Gantmacher, F.R., *The Theory of Matrices*, Chelsea Publishing Co., New York, 1960.

III. THE DYNAMIC SIMPLEX METHOD: A BASIS FACTORIZATION APPROACH

1. Introduction

In this part, separate operations and the general scheme of the dynamic simplex-method will be described. An illustrative numerical example and the theoretical evaluation of the algorithm are given. In conclusion, we consider briefly important extensions of the algorithm (non-negative state constraints, time delays in state and control variables, etc.).

For convenience, we repeat the statement of the problem below [1].

Problem 1.1: Find a control $u = \{u(0),\dots,u(t-1)\}$ and a corresponding trajectory $x = \{x(0),\dots,x(T)\}$ satisfying the state equations

$$x(t+1) = A(t)x(t) + B(t)u(t) \tag{1.1}$$

with initial condition

$$x(0) = x^0 \tag{1.2}$$

and constraints

$$G(t)x(t) + B(t)u(t) = f(t) \tag{1.3}$$

$$u(t) \geq 0 \tag{1.4}$$

which maximize the objective function

$$J_1(u) = a(T)x(T) \quad . \tag{1.5}$$

Here we use the same notations as in Parts I and II.

Problem 1.1 can be considered as some "large" linear programming problem with constraints (1.1) to (1.4). The constraint matrix of Problem 1.1 has a staircase structure and dimension $(r+n)T \times (m+n)T$; decision variables are $\{u,x\} = \{u_k(t), x_i(t+1)$ $(k=1,\dots,r;\ i=1,\dots,n;\ t=0,\dots,T-1)\}$.

We shall denote a basic feasible solution of Problem 1.1 by $\{u_B, x\}$ (the free variables x are always in a basis). Evidently, u_B is a basic feasible control in the sense of Definition 1.2 [1].

2. Basis Factorization Approach

The method which was considered in [1], can be interpreted as some basis factorization approach to Problem 1.1's solution. Below we describe the method in these terms.

We need the following assertion.

Theorem 2.1: [2]: *Let a non-singular square matrix* F *be partitioned into blocks*

$$F = \begin{array}{cc} \begin{array}{cc} \overbrace{}^{m} & \overbrace{}^{n} \end{array} & \\ \left[\begin{array}{c:c} H & P \\ \hdashline Q & R \end{array}\right] & \begin{array}{l} \}m \\ \}n \end{array} \end{array} \tag{2.1}$$

where H is a non-singular matrix.

Then F *is represented as the product of upper and lower triangular matrices in the form*

$$F = \bar{F} \cdot U = \left[\begin{array}{c:c} H & O \\ \hdashline Q & C \end{array}\right] \cdot \begin{array}{cc} \begin{array}{cc} \overbrace{}^{m} & \overbrace{}^{n} \end{array} & \\ \left[\begin{array}{c:c} I_m & \phi \\ \hdashline O & I_n \end{array}\right] & \begin{array}{l} \}m \\ \}n \end{array} \end{array} , \tag{2.2}$$

where

$$C = R - QH^{-1}P \quad , \qquad |C| \neq 0 \quad , \qquad \phi = H^{-1}P \quad , \tag{2.3}$$

I_m *and* I_n *are the identity matrices of appropriate dimensions; the inverse of each of the factors is readily obtained and their product yields the inverse of* F:

$$F^{-1} = \left[\begin{array}{c:c} H^{-1}+H^{-1}PC^{-1}QH^{-1} & -H^{-1}PC^{-1} \\ \hdashline -C^{-1}QH^{-1} & C^{-1} \end{array}\right] . \tag{2.4}$$

Theorem 2.1 is not stated in [2] in explicit form, but directly follows from results given in [2].

We now apply the theorem to Problem 1.1. The basis matrix $\bar{B}$ of Problem 1.1 has the same structure as the constraint matrix:

$$\bar{B} = \begin{bmatrix} D_B(0) & & & & & & \\ B_B(0) & -I & & & & & \\ & G(1) & D_B(1) & & & & \\ & A(1) & B_B(1) & -I & & & \\ & & & \ddots & & & \\ & & & & -I & & \\ & & & & G(T-1) & D_B(T-1) & \\ & & & & A(T-1) & B_B(T-1) & -I \end{bmatrix} \tag{2.5}$$

where I is the identity matrix of dimension $n \times n$, $D_B(t)$ and $B_B(t)$ are submatrices, formed by basic columns of the constraint matrix.

As the rows of $D_B(0)$ are linearly independent, one can choose m linearly independent columns in the matrix $D_B(0)$. These columns generate the matrix $\hat{D}_{0B}(0)$.

By column permutation, we can transform the matrix $D_B(0)$ and obtain $D_B(0) = [\hat{D}_{0B}(0);\hat{D}_{1B}(0)]$, where $\hat{D}_{1B}(0)$ is the submatrix, consisting of the columns of the matrix $D_B(0)$ which are not in the matrix $\hat{D}_{0B}(0)$.

The column permutation of the matrix $D_B(0)$ implies the corresponding partition of the matrix $B_B(0)$: $B_B(0) = [\hat{B}_{0B}(0);\hat{B}_{1B}(0)]$.

In accordance with Theorem 2.1, one can show that the matrix $\bar{B}$ is expressed as

$$\bar{B} = \bar{B}_0 U_0 \tag{2.6}$$

where U_0 is the upper triangular matrix whose dimensions conform with those of $\bar{B}$.

$$U_0 = \begin{bmatrix} 1 & & & & \\ & \cdot & & \Phi_B(0) & \\ & & \ddots & & \\ & 0 & & \cdot & \\ & & & & 1 \end{bmatrix}$$

In the matrix U_0, the dimension and location of the matrix

$$\Phi_B(0) = \hat{D}_{0B}^{-1}(0)\hat{D}_{1B}(0)$$

coincide with the dimension and location of the matrix $\hat{D}_{1B}(0)$ in $\bar{B}$. The matrix $\bar{B}_0$ is obtained from the matrix $\bar{B}$ through replacement $[\hat{D}_{0B}(0);\hat{D}_{1B}(0)]$ by $[\hat{D}_{0B}(0);0]$ and $\hat{B}_{1B}(0)$ by $B_B^1(0) = \hat{B}_{1B}(0) - \hat{B}_{0B}(0)\Phi_B(0)$.

In the matrix $\bar{B}_0$, we permute the submatrix -I and the submatrix $B_B^1(0)$. Then we permute the submatrices G(1) and A(1) in the matrix $\bar{B}_0$ and the submatrix $\Phi_B(0)$ in the matrix U_0 respectively.

By analogy with (2.6), we can write that $\bar{B}_0 = \bar{B}_1 V_0$, where V_0 is the upper triangular matrix of the matrix $\bar{B}_0$ dimension:

$$V_0 = \begin{bmatrix} 1 & & & & \\ & \cdot & & -B_B^1(0) & \\ & & \ddots & & \\ & 0 & & \cdot & \\ & & & & 1 \end{bmatrix} ,$$

$$B_B^1(0) = \hat{B}_{1B}(0) - \hat{B}_{0B}(0)\Phi_B(0) \quad ,$$

and

$$\bar{B}_1 = \left[\begin{array}{ccccccccccc}
\hat{D}_{0B}(0) & & & & & & & & & & \\
\hat{B}_{0B}(0) & -I & & & & & & & & & \\
 & G(1) & G(1)B_B^1(0) & D_B(1) & & & & & & & \\
 & A(1) & A(1)B_B^1(0) & B_B(1) & -I & & & & & & \\
 & & & & G(2) & D_B(2) & & & & & \\
 & & & & A(2) & B_B(2) & -I & & & & \\
 & & & & & & & \ddots & & & \\
 & & & & & & & & G(T-1) & D_B(T-1) & \\
 & & & & & & & & A(T-1) & B_B(T-1) & -I
\end{array}\right] .$$

The dimension and location of the matrix $-B_B^1(0)$ in V_0 coincide with the dimension and location of the matrix $B_B^1(0)$ in $\bar{B}_0$. The matrix $\bar{B}_1$ is obtained from $\bar{B}_0$ by the replacement of submatrices

$$\begin{array}{rcl}
[-I : B_B^1(0)] & \text{by} & [-I : 0] \quad , \\
{[G(1) : 0 : D_B(1)]} & \text{by} & [G(1) : G(1)B_B^1(0) : D_B(1)] \quad , \\
{[A(1) : 0 : B_B(1)]} & \text{by} & [A(1) : A(1)B_B^1(0) : B_B(1)] \quad .
\end{array}$$

In accordance with Theorem 2.1, a matrix, obtained from the matrix $\bar{B}_1$ by cutting out the rows coinciding with the rows of submatrices $\hat{D}_{0B}(0)$ and $\hat{B}_{0B}(0)$ and by cutting out the columns coinciding with the columns of submatrices $\hat{D}_{0B}(0)$ and $G(1)$, is nonsingular. Consequently, the rows of the matrix

$$[G(1)B_B^1(0) : D_B(1)]$$

are linearly independent, and by column permutation, this matrix can be reduced to the form

$$[G(1)B_B^1(0) : D_B(1)] = [\hat{D}_{0B}(1) : \hat{D}_{1B}(1)] \quad ,$$

where the matrix $\hat{D}_{0B}(1)$ is nonsingular and the matrix $\hat{D}_{1B}(1)$ is generated by columns $[G(1)B_B^1(0) : D(1)]$, which are not in the matrix $\hat{D}_{0B}(1)$.

The matrices

$$[A(1)B_B^1(0) : B_B(1)] = [\hat{B}_{0B}(1) : \hat{B}_{1B}(1)]$$

and $\Phi_B(0)$ in matrix U_0, as well as the matrix $-B_B^{-1}(0)$ in the matrix V_0, are partitioned similarly.

Proceeding in a similar way, we obtain

$$\bar{B} = B^* V_{T-2} U_{T-2} \;\cdots\cdots\; V_0 U_0 = B^* U \quad , \tag{2.7}$$

where

$$B^* = \begin{bmatrix} \hat{D}_{0B}(0) & & & & & & \\ \hat{B}_{0B}(0) & -I & & & & & \\ & G(1) & \hat{D}_{0B}(1) & & & & \\ & A(1) & \hat{B}_{0B}(1) & -I & & & \\ & & & & \ddots & & \\ & & & & & G(T-1)\hat{D}_{0B}(T-1) & \\ & & & & & A(T-1)\hat{B}_{0B}(T-1) & -I \end{bmatrix} \tag{2.7a}$$

where $\hat{D}_{0B}(t)\,(t = 0,\ldots,T-1)$ is a square non-singular matrix of dimension $m \times m$ and is formed either by columns of the matrix $D(t)$ or by some columns of matrices $D(\tau)\,(\tau = 0,\ldots,t-1)$, which are recomputed to step t during factorization process. Evidently, the matrices $\hat{D}_{0B}(t)$ $(t = 0,1,\ldots,T-1)$, obtained in such a way, coincide with the local bases, which were defined in [1].

The matrices U_t and V_t $(t = 0,1,\ldots,T-2)$ are

$$U_t = \begin{bmatrix} 1 & & & & 0 & & \\ & 1 & & & & & \\ & & \cdot\; 0 \ldots \Phi_0^{t+1}(t) \ldots 0 \ldots \Phi_i^{j}(t) \ldots 0 \ldots \Phi_t^{T-1}(t) \ldots 0 & & & & \\ & & & \ddots & & 0 & \\ & 0 & & & \ddots & & \\ & & & & & 1 & \\ & & & & & & 1 \end{bmatrix}$$

and

$$V_t = \begin{bmatrix} 1 & & & & 0 & & \\ & 1 & & & & & \\ & & \cdot\; 0 \ldots -B_0^{t+1}(t) \ldots 0 \ldots -B_i^{j}(t) \ldots 0 \ldots -B_t^{T-1}(t) \ldots 0 & & & & \\ & & & \ddots & & 0 & \\ & 0 & & & \ddots & & \\ & & & & & 1 & \\ & & & & & & 1 \end{bmatrix}$$

where $\Phi_i^j(t)$ and $-B_i^j(t)$ correspond to those basic control variables $u_B(i)$, which enter local basis $\hat{D}_{0B}(j)$. Location of rows of submatrices $\Phi_i^j(t)$ and $-B_i^j(t)$ correspond to the location of rows of submatrices $\hat{D}_{0B}(t)$ and $\hat{B}_{0B}(t)$ in B^*.

We denote the non-zero columns in the right corners of the matrices U_t and V_t by $\Phi_B(t)$ and $B_B^1(t)$:

$$\Phi_B(t) = [\Phi_0^{t+1}(t) \ \dots \ \Phi_i^j(t) \ \dots \ \Phi_t^{T-1}(t)]$$

$$B_B^1(t) = [-B_0^{t+1}(t) \ \dots \ -B_i^j(t) \ \dots \ -B_t^{T-1}(t)] \quad .$$

By construction, these matrices are defined from

$$\Phi_B(t) = \hat{D}_{0B}^{-1}(t)\hat{D}_{1B}(t) \quad , \tag{2.8}$$

$$B_B^1(t) = \hat{B}_{1B}(t) - \hat{B}_{0B}(t)\Phi_B(t) \quad . \tag{2.9}$$

One can see that these matrices conform with the matrices defined by fomulas (2.12) and (2.17) in [1].

Taking into account the permutation of basis columns in the factorization process, we can write the basic variables as

$$\{u_B, x\} = \{\hat{u}_{0B}(0), x(1), \hat{u}_{0B}(1), \dots, \hat{u}_{0B}(T-1), x(T)\} \quad ,$$

where vector $\hat{u}_{0B}(t)$ corresponds to matrix $\hat{D}_{0B}(t)$ $(t = 0,1,\dots,T-1)$.

At each simplex iteration, it is necessary to solve three system of linear equations for:

(1) determination of a basic solution;

(2) computation of coefficients $\{v,y\}$ which are the representation of the incoming vector

$$Y_j(t_1) = (0,\dots,0,d_j^T(t_1),b_j^T(t_1),0,\dots,0)^T$$

in terms of the basis;

(3) determination of the simplex-multipliers.

Now we describe these procedures for factorized representation of the basis. We single out the following procedures: the primal solution, the dual solution; pricing and updating.

3. Primal Solution

Vector $X = (u_B, x)$ is calculated from the solution of the system

$$\bar{B}X = B^*UX = B^*V_{T-2} \dots U_0X = b \quad , \tag{3.1}$$

where b is the constraint vector of Problem 1.1.

Denote

$$X^* = UX \quad ;$$

then the calculation of the vector X reduces to subsequent solution of two systems of linear equations in forward and backward runs:

$$B^*X^* = b \quad , \tag{3.2}$$

$$UX = X^* \quad . \tag{3.3}$$

The solution of (3.2) is determined by recurrent formulas:

$$\begin{aligned} \hat{u}^*_{0B}(t) &= \hat{D}^{-1}_{0B}(t)(f(t) - G(t)x^*(t)) \qquad (t = 0,\dots,T-1) \quad , \\ x^*(t+1) &= A(t)x^*(t) + \hat{B}_{0B}(t)\hat{u}^*_{0B}(t) \qquad (t = 0,\dots,T-1) \quad , \\ x^*(0) &= x(0) \quad . \end{aligned} \tag{3.4}$$

The system (3.3), considering (2.3), can be written as

$$X = U_0^{-1} \dots V_{T-2}^{-1}X^* \quad .$$

It is easy to see that the matrices U_t^{-1} and V_t^{-1} are obtained from the matrices U_t and V_t by simply changing the signs of the elements which are above the main diagonal. Therefore the solution of the system (3.3) reduces to the recurrent formulas:

$$\begin{aligned}
x(T) &= x^*(T) \quad , \\
u(T-1) &= u^*(T-1) \quad , \\
x(T) &= x^*(t) + \sum_{i=0}^{t-1} \sum_{j=t}^{T-1} [B_i^j(t) : 0]\hat{u}_{0B}(j) \ , \quad (t=T-1,\ldots,1) \\
\hat{u}_{0B}(t) &= \hat{u}^*_{0B}(t) - \sum_{i=0}^{t} \sum_{j=t+1}^{T-1} [\Phi_i^j(t) : 0]\hat{u}_{0B}(j) , (t=T-2,\ldots,0) \quad .
\end{aligned} \tag{3.5}$$

Here the notation $[B_i^j(t) : 0]$ and $[\Phi_i^j(t) : 0]$ denote that the matrices $B_i^j(t)$ and $\Phi_i^j(T)$ are augmented by zeros, if necessary, so that the matrices conform with multiplying.

The coefficients

$$\bar{Y}_j(t_1) = (\hat{v}_{0B}(0), y(1), \hat{v}_{0B}(1), \ldots, y(T)) \quad ,$$

which represent the vector $Y_j(t_1)$ in terms of the basis, are calculated from the solution of the system

$$\bar{B}\bar{Y}_j(t_1) = Y_j(t_1) \quad .$$

On the forward run, we find the vector sequence (v^*,y^*):

$$\begin{aligned}
\hat{v}^*_{0B}(t) &= 0 \quad , \\
y^*(t+1) &= 0 \quad , \qquad (t=0,\ldots,t_1-1) \quad , \\
\hat{v}^*_{0B}(t_1) &= \hat{D}_{0B}^{-1}(t_1)d_j(t_1) \quad , \\
y^*(t_1+1) &= \hat{B}_{0B}(t_1)\hat{v}^*_{0B}(t_1) - b_j(t_1) \quad , \\
\hat{v}^*_{0B}(t) &= -\hat{D}_{0B}^{-1}(t)G(t)y^*(t) \quad , \\
y^*(t+1) &= A(t)y^*(t) + \hat{B}_{0B}(t)\hat{v}^*_{0B}(t) , \quad (t=t_1+1,\ldots,T-1)
\end{aligned} \tag{3.6}$$

On the backward run, we find vector sequence (v,y);

$$
\begin{aligned}
y(T) &= y^*(T) \quad , \\
\hat{v}_{0B}(T-1) &= v^*_{0B}(T-1) \quad , \\
y(t) &= y^*(t) + \sum_{i=0}^{t-1} \sum_{j=t}^{T-1} [B_i^j(t) : 0]\hat{v}_{0B}(j), \qquad (t = T-1,\dots,1) \quad , \qquad (3.7) \\
\hat{v}_{0B}(t) &= \hat{v}^*_{0B}(t) - \sum_{i=0}^{t} \sum_{j=t+1}^{T-1} [\Phi_i^j(t) : 0]\hat{v}_{0B}(j) \qquad (t = T-2,\dots,0) \quad .
\end{aligned}
$$

For given sequences $\hat{u}$ and $\hat{v}$, the pair of indices (ℓ,t_2) which correspond to the outgoing vector, is defined by

$$
\theta_0 = \min_{\substack{(i,\tau) \\ \hat{v}_{0i}(\tau)>0}} \frac{\hat{u}_{0i}(\tau)}{\hat{v}_{0i}(\tau)} = \frac{\hat{u}_{0\ell}(t_2)}{\hat{v}_{0\ell}(t_2)} \quad . \qquad (3.8)
$$

4. Dual Solution

We define $(n+m)T$-vector $\pi = \{\lambda,p\}$ as

$$
c_B = \pi\bar{B} \qquad (4.1)
$$

where $\bar{B}$ is a basis matrix (2.5) and $c_B = \{0,\dots,0,a(T)\}$. From (4.1) and representation (2.7) of the basis matrix $\bar{B}$, we can calculate the simplex-multipliers $\{\lambda,p\} = \{\lambda(0),p(1),\dots,\lambda(T-1),p(T)\}$ in a similar way using the same matrices $\hat{B}_{0B}(t)$ and $\hat{D}^{-1}_{0B}(t)$:

$$
\begin{aligned}
p(T) &= a(T) \quad , \\
\lambda(t) &= p(t+1)\hat{B}_{0B}(t)\hat{D}^{-1}_{0B}(t), \qquad (t = T-1,\dots,0) \qquad (4.2) \\
p(t) &= p(t+1)A(t) - \lambda(t)G(t), \quad (t = T-1,\dots,1) \quad .
\end{aligned}
$$

One can see that the formulas (3.4) to (3.7) are the explicit expression of Procedure 1 [1] for determination of basic variables and coefficients, expressing a column not in the basis by the basis columns. The formulas (4.2) for determination of simplex-multipliers coincide with the formulas of Procedure 3 [1].

5. Pricing

The pricing procedure is now constructed straightforwardly. To price out a vector $d_j(t)$ which is not in the basis, we use formulas [1]:

$$\Delta_j(t) = \lambda(t)d_j(t) - p(t+1)b_j(t) \tag{5.1}$$

where the simplex-multipliers $\lambda(t)$ and $p(t+1)$ are defined from (4.2).

It should be noted that the method requires only partial pricing: that is, to determine $\lambda(t_1)$ and $p(t_1+1)$, which are needed for pricing out the nonbasic components of vector $u(t_1)$, one has to calculate vectors $\lambda(t)$ and $p(t+1)$ only for $t = T-1, T-2, \ldots, t_1+1$. These computations require only a part of the basis inverse representation, in particular, only a few of the local bases. In a standard approach it is generally not possible to compute part of the components of the simplex-multiplier vector without computing the whole vector.

6. Updating

The pricing procedure of computing the values $\lambda_j(t)$ for vectors $d_j(t)$, $(j,t) \in I_N(u)$, which are not in the basis allows us to define the vector to be introduced into the basis and the vector to be removed from the basis.

Let $d_j(t_1)$ be the ingoing column vector and $\hat{d}_{0\ell}(t_2)$ be the outgoing column vector. Here $d_j(t_1)$ is the j-th nonbasic column of the matrix $D(t_1)$ and $\hat{d}_{0\ell}(t_2)$ is the ℓ-th column of the matrix $\hat{D}_{0B}(t_2)$, $0 \le t_1, t_2 \le T-1$.

Replacing the vector $d_j(t_1)$ by the vector $\hat{d}_{0\ell}(t_2)$ implies the updating of the old system of local basis $\{\hat{D}_{0B}(t)\}$ by a new system of local bases $\{\hat{D}_{0B}(t)\}'$.

As in the case of the static simplex method, the updating procedure must be done efficiently as it constitutes the main effort of each iterative cycle of the algorithm.

In the dynamic simplex method, we operate with the system of inverses $\{\hat{D}_{0B}^{-1}(t)(t=0,1,\ldots,T-1)\}$ of local bases. Hence the efficiency of the method will be directly defined by the updating scheme of the inverses $\{\hat{D}_{0B}^{-1}(t)\}$.

The main complications of the updating procedure in the dynamic case is the fact that, first, the updating of a local basis at step t can change the subsequent local bases $\hat{D}_{0B}(\tau)$ $(\tau = t+1,\ldots,T-1)$ and that, second, the outgoing vector $\hat{d}_{0\ell}(t_2)$ may belong to the local basis $\hat{D}_{0B}(t_2)$ at another step t_2, $t_2 \neq t_1$.

The theorem below gives a sufficient condition when the replacement of a basis column in a local basis $\hat{D}_{0B}(t)$ does not change the other local bases.

Theorem 6.1: The replacement of the i*-th column in a local basis* $\hat{D}_{0B}(t)$ *does not change the other local bases, if the* i*-th row of the matrix* $\Phi_B(t)$*, where* $\Phi_B(t)$ *defined by (2.8), vanishes.*

Proof: When we replace the i-th column in the matrix $\hat{D}_{0B}(t)$, then in accordance with (2.7), the updating of the matrix $\Phi_B(t)$ will be similar to the updating of the inverse $\hat{D}_{0B}^{-1}(t)$, that is, the i-th pivot row of the matrix is added to the other row with some coefficients [3].

Therefore, if the i-th row of the matrix $\Phi_B(t)$ is zero, the matrix $\Phi_B(t)$ will not change. In accordance with (2.9), the matrix $B_B^1(t)$ does not change either. Considering the construction of matrices $\hat{D}_{0B}(t)$ at next steps, we find that all subsequent local bases $\hat{D}_{0B}(t)$ $(\tau = t+1, t+2,\ldots,T-1)$ also do not change.

Consequence 6.1: If an element $\phi_{ij}(t)$ of the matrix $\Phi_B(t)$ is zero, then the replacement of the i-th column in the local basis $\hat{D}_{0B}(t)$ does not change the j-th column in the matrix $B_B^1(t)$.

Now let us consider the interchange of the ℓ-th column of local basis $\hat{D}_{0B}(t)$ with a column of local basis $\hat{D}_{0B}(t+1)$ and find how the inverses $\hat{D}_{0B}^{-1}(\tau)$ and matrices $\Phi_B(\tau)$, $B_B^1(\tau)$ $(\tau = t,\ldots,T-1)$ are updated at this interchange. For this, we need the following theorem.

Theorem 6.2: Let the k*-th column of submatrix* $\begin{bmatrix} H \\ Q \end{bmatrix}$ *of the matrix* F *in (2.1) be interchanged with the* ℓ*-th column of submatrix* $\begin{bmatrix} P \\ R \end{bmatrix}$ *and let the element* $\phi_{k\ell}$ *of the matrix* $\Phi = H^{-1}P$ *be not zero (pivoting element). Then, denoting the updated submatrices in* F *as* H', Q', P' *and* R', *the following relations hold:*

$$\text{(i)} \quad (H^{-1})' = E_k H^{-1} \tag{6.1}$$

where E_k *is an elementary* $(m \times m)$ *column matrix with elements of the non-zero k-th column:*

$$\eta_i = -\frac{\phi_{i\ell}}{\phi_{k\ell}} \quad (i = 1,\ldots,n,\ i \neq k)$$

$$\eta_k = \frac{1}{\phi_{k\ell}}$$

$$\text{(ii)} \quad \Phi_i' = E_k\Phi_i \quad i \neq \ell,\ \Phi_\ell' = [\eta_1,\ldots,\eta_m]^T \ , \tag{6.2}$$

$$(C^{-1})' = E_\ell C^{-1} \ , \tag{6.3}$$

where Φ_i *is the* i*-th column of* Φ, E_ℓ *is an elementary* $(n \times n)$ *row matrix with elements of the non-zero* ℓ*-th row equal to* $-\phi_{\ell i}$ $(i = 1,\ldots,n)$*;*

$$\text{(iii)} \quad C' = CE_\ell^{-1} \tag{6.4}$$

where E_ℓ^{-1} *is an elementary* $(n \times n)$ *row matrix with elements of the non-zero* ℓ*-th row:*

$$\eta_i = -\frac{\phi_{ki}}{\phi_{k\ell}} \qquad (i = 1,\ldots,n; i \neq \ell)$$

$$\eta_\ell = -\frac{1}{\phi_{k\ell}}$$

Proof: Formulas (6.1) and (6.2) are the basis updating formulas in the simplex method [3]. Now, to prove (6.3): the column permutations of the matrix F can be written as a matrix product $\hat{F} = FT$, where

$$T = \begin{array}{c} \left[\begin{array}{ccccccc} 1 & & & \vdots & & & \\ & \ddots & & \vdots & & & \\ & & 0 \cdots & \vdots & \cdots 1 & & \\ & & \vdots \ \ddots & \vdots & \vdots & & \\ \cdots & \cdots & \vdots \cdots 1 & \vdots & \cdots \vdots & \cdots & \\ & & \vdots & \vdots\ 1 & \vdots & & \\ & & \vdots & \vdots & \ddots \ \vdots & & \\ & & 1 \cdots & \vdots & \cdots 0 & & \\ & & & \vdots & & \ddots & \\ & & & \vdots & & & 1 \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c}1\\1\\1\\1\end{array}}\right\} m \\ \\ \left.\vphantom{\begin{array}{c}1\\1\\1\\1\end{array}}\right\} n \end{array} \\ \qquad\qquad \underbrace{\qquad}_{k} \qquad\quad \underbrace{\qquad}_{\ell} \end{array} .$$

As $T^{-1} = T$, then $\hat{F}^{-1} = TF^{-1}$. Taking into account the partitioning of the matrices and using Theorem 2.1, we obtain

$$\hat{C}^1 = \underbrace{\begin{bmatrix} 0 \ldots 0 \ldots 0 \\ \vdots \quad\ \vdots \quad\ \vdots \\ 0 \ldots 1 \ldots 0 \\ \vdots \quad\ \vdots \quad\ \vdots \\ 0 \ldots 0 \ldots 0 \end{bmatrix}}_{k} H^{-1}PC^{-1} + \underbrace{\begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 0 & \\ & & \vdots \ddots & \\ & & & 1 \end{bmatrix}}_{\ell} C^{-1}$$

$$= -\begin{bmatrix} 0 \ldots 0 \ldots 0 \\ \vdots \quad\ \vdots \quad\ \vdots \\ 0 \ldots 1 \ldots 0 \\ \vdots \quad\ \vdots \quad\ \vdots \\ 0 \ldots 0 \ldots 0 \end{bmatrix} \Phi C^{-1} + \begin{bmatrix} 1 & & \\ & \ddots & \\ & 0 & \\ & & \ddots \\ & & & 1 \end{bmatrix} C^{-1} = E_\ell C^{-1} \quad .$$

The relation (6.4) follows directly from (6.3). This completes the proof of the theorem.

Now let $\phi_{\ell q}(t) \neq 0$ be the pivoting element of the matrix $\Phi_B(t)$, which correspond to the q-th component of the vector $\hat{u}_{0B}(t+1)$. According to Theorem 6.2, at the interchange of the ℓ-th column of the matrix $\hat{D}_{0B}(t)$ with the q-th column of the matrix $\hat{D}_{1B}(t)$, the inverse $\hat{D}_{0B}^{-1}(t)$ is updated by premultiplying on the elementary matrix. The elementary matrix has dimension $m \times m$ and differs from the identity matrix by the ℓ-th column with components [3]:

$$\eta_i = -\frac{\phi_{iq}(t)}{\phi_{\ell q}(t)} \qquad (i = 1,\ldots,m;\ i \neq \ell) \quad ;$$

$$\eta_i = \frac{1}{\phi_{\ell q}(t)} \qquad (i = \ell) \quad .$$

The column permutation in matrices $\hat{B}_{0B}(t)$ and $\hat{B}_{1B}(t)$ is carried out in a similar way. The matrix $B_B^1(t)$ is updated according to Theorem 6.2 as

$$[B_B^1(t)]' = B_B^1(t)E_q \quad , \tag{6.5}$$

where E_q is an elementary row matrix, which differs from the identity matrix by the q-th row with components

$$\xi_i(t) = \frac{\phi_{\ell i}(t)}{\phi_{\ell q}(t)} \quad , \qquad i \neq q \quad ;$$

$$\xi_i(t) = -\frac{1}{\phi_{\ell q}(t)} \quad , \qquad i = q \quad .$$

Define now the updating of the inverses $\hat{D}_{0B}^{-1}(\tau)$ $(\tau = t+1,\ldots,T-1)$.

<u>*Theorem 6.3:*</u> *Let* $\phi_{\ell q}(t) \neq 0$ *be the pivoting element of the matrix* $\Phi_B(t)$ *(which corresponds to the q-th component of vector* $\hat{u}_{0B}(t+1)$*).*

Then at the interchange of the ℓ*-th column of* $\hat{D}_{0B}(t)$ *with the* q*-th column of* $\hat{D}_{0B}(t+1)$ *the following relations hold:*

$$[\hat{D}_{0B}^{-1}(t+1)]' = L_q^{-1}\hat{D}_{0B}^{-1}(t+1) \tag{6.6}$$

$$[\hat{B}_{0B}(t+1)]' = \hat{B}_{0B}(t+1)L_q \tag{6.7}$$

$$[\Phi_B(t+1)]' = L_q^{-1}N_q + L_q^{-1}\Phi_B(t+1) \tag{6.8}$$

where L_q *is an elementary row* $(m \times m)$ *matrix, which differs from the unit matrix by the* q*-th row, and* N_q *is a matrix, which differs from the zero matrix by the* q*-th row.*

The matrix $B_B^1(t+1)$ *is not changed at this permutation, neither are all the subsequent local bases* $\hat{D}_{0B}(\tau)$ *and matrices* $\Phi_B(\tau)$, $B_B^1(\tau)$ $(\tau = t+2,\ldots,T-1)$.

Proof: Taking into account the structure of the matrices $\hat{D}\ (t+1)$ and $\hat{B}_B(t+1)$, we can write (after column permutations):

$$\begin{aligned}\hat{D}_B(t+1) &= [G(t+1)B_B^1(t+1);\ D_B(t+1)] \\ &= [\hat{D}_{0B}(t+1);\ \hat{D}_{1B}(t+1)]\end{aligned} \tag{6.9}$$

and

$$\begin{aligned}\hat{B}_B(t+1) &= [A(t+1)B_B^1(t+1);\ B_B(t+1)] \\ &= [\hat{B}_{0B}(t+1),\ \hat{B}_{1B}(t+1)] \quad .\end{aligned} \tag{6.10}$$

Considering (6.9) and (6.10), we obtain

$$\begin{bmatrix}[\hat{D}_{0B}(t+1)]'[\hat{D}_{1B}(t+1)]' \\ [\hat{B}_{0B}(t+1)]'[\hat{B}_{1B}(t+1)]'\end{bmatrix} = \begin{bmatrix}\hat{D}_{0B}(t+1) & \hat{D}_{1B}(t+1) \\ \hat{B}_{0B}(t+1) & \hat{B}_{1B}(t+1)\end{bmatrix}\begin{bmatrix}L_q & N_q \\ 0 & I\end{bmatrix} \tag{6.11}$$

Here $[\hat{D}_B(t+1)]'$, $[\hat{B}_B(t+1)]'$ are the updated matrices corresponding to the new basis; L_q is the elementary row $(m \times m)$ matrix; N_q is the $(m \times k)$ matrix; I is the identity matrix of dimension $(k \times k)$.

The right matrix in (6.11) is built up as follows: the matrix E_q in (6.5) is enlarged up to dimension $(m+k) \times (m+k)$ in such a way that in the added part the main diagonal contains units and all the remaining added elements are zero; then the elements of the q-th row are permuted in accordance with the columns permutations of the matrix $\hat{D}_B(t+1)$, when it is partitioned on the matrices $\hat{D}_{0B}(t+1)$ and $\hat{D}_{1B}(t+1)$.

Multiplying the right-hand matrices in (6.11) and taking into account their partitioning, we obtain (6.6) and (6.7). Besides,

$$\begin{aligned}
[\hat{D}_{1B}(t+1)]' &= \hat{D}_{0B}(t+1)N_q + \hat{D}_{1B}(t+1) \\
[\hat{B}_{1B}(t+1)]' &= \hat{B}_{0B}(t+1)N_q + \hat{B}_{1B}(t+1) \quad .
\end{aligned} \qquad (6.12).$$

According to (2.9), we have

$$[B_B^1(t+1)]' = [\hat{B}_{1B}(t+1)' - [\hat{B}_{0B}(t+1)]'[\hat{D}_{0B}^{-1}(t+1)]'[\hat{D}_{1B}(t+1)]' \qquad (6.13)$$

Substituting (6.6) and (6.7) into (6.13), we obtain

$$\begin{aligned}
[B_B^1(t+1)' &= \hat{B}_{0B}(t+1)N_q + \hat{B}_{1B}(t+1) - \hat{B}_{0B}(t+1)L_q L_q^{-1} \\
&\quad \times \hat{D}_{0B}^{-1}(t+1)\ (\hat{D}_{0B}(t+1)N_q + \hat{D}_{1B}(t+1)) \\
&= \hat{B}_{1B}(t+1) - \hat{B}_{0B}(t+1)\hat{D}_{0B}^{-1}(t+1)\hat{D}_{1B}(t+1) \\
&= B_B^1(t+1) \quad .
\end{aligned}$$

The matrix $B_B^1(t+1)$ is not changed, therefore all the subsequent local bases $\hat{D}_{0B}(\tau)$ and matrices $\Phi_B(\tau)$, $B_B^1(\tau)$ $(\tau = T+2, \ldots, T-1)$ are not changed.

Finally, taking into account (6.6) and (6.12), we obtain (6.8):

$$[\Phi_B(t+1)]' = [\hat{D}_{0B}^{-1}(t+1)]'[\hat{D}_{1B}(t+1)]' = L_q^{-1}N_q + L_q^{-1}\Phi_B(t+1) \quad .$$

This procedure we shall call *the interchange of the ℓ-th column of the matrix* $\hat{D}_{0B}(t)$ *with the q-th column of the matrix* $\hat{D}_{0B}(t+1)$.

Now let us consider the interchange of the ℓ-th column of the matrix $\hat{D}_{0B}(t)$ with some column of the matrix $\hat{D}_{0B}(t^*)$, $t^* > t+1$.

In the ℓ row of the matrix $\Phi_B(t)$, let the first non-zero element $\phi_{\ell q}(t)$ correspond to the basic variable, which is recomputed to the local basis $\hat{D}_{0B}(t^*)$, and all elements $\phi_{\ell i}(t)$, corresponding to the variables which are recomputed to local bases $\hat{D}_{0B}(\tau)$, $t < \tau < t^*$, equal to zero. Now we partition the matrices $\Phi_B(t)$ and $\hat{B}_{1B}(t)$ into two parts:

$$\begin{aligned} \Phi_B(t) &= [\Phi_1(t);\ \Phi_2(t)] \quad ; \\ \hat{B}_{1B}(t) &= [\hat{B}_{11}(t);\ \hat{B}_{12}(t)] \quad ; \\ B_B^1(t) &= [B_1^1(t);\ B_2^1(t)] \quad . \end{aligned} \tag{6.14}$$

Let the columns corresponding to the variables which are recomputed into the local bases $\hat{D}_{0B}(\tau)$, $t < \tau < t^*$, enter the matrix $\Phi_1(t)$ $(\hat{B}_{11}(t))$, $B_1^1(t)$), and the remaining columns enter the matrix $\Phi_2(t)$ $(\hat{B}_{12}(t), B_2^1(t))$.

Then, in accordance with (6.5) and consequence 6.1, the matrix $B_1^1(t)$ does not change at the interchange of the ℓ-th column of the matrix $\hat{D}_{0B}(t)$ with the q-th column of the matrix $\hat{D}_{1B}(t)$. The matrix $B_2^1(t)$, which is defined from (6.14) is transformed in accordance with the formula

$$[B_2^1(t)]' = B_2^1(t)E_k \quad ,$$

where k is the number of the column of the matrix $\Phi_2(t)$ which contains the element $\Phi_{\ell q}(t)$. The order of matrix E_k is equal to the number of columns of matrix $B_2^1(t)$.

Let the k-th column of matrix $\Phi_2(t)$ correspond to the k-th component (6.9), (6.10), the columns of matrices

$$G(t+1)B_2^1(t) \text{ and } A(t+1)B_2^1(t)$$

do not enter the matrices

$$\hat{D}_{0B}(t+1) \text{ and } \hat{B}_{0B}(t+1) \quad .$$

Therefore the matrices $\hat{D}_{0B}(t+1)$, $\hat{B}_{0B}(t+1)$ do not change.

Let us partition the matrices $\Phi_B(t+1)$, $B_B^1(t+1)$ and $\hat{B}_{1B}(t+1)$ into two submatrices

$$\begin{aligned} \Phi_B(t+1) &= [\Phi_1(t+1);\ \Phi_2(t+1)] \quad ; \\ B_B^1(t+1) &= [B_1^1(t+1);\ B_2^1(t+1)] \quad ; \\ B_{1B}^1(t+1) &= [\hat{B}_{11}(t+1);\ \hat{B}_{12}(t+1)] \quad . \end{aligned}$$

The columns of the matrix $\Phi_B(t+1)$, which correspond to the same basic elements as the columns of the matrix $\Phi_2(t)$, enter the matrix $\Phi_2(t+1)$.

In accordance with the partitioning, the matrices $\Phi_1(t+1)$ and $\hat{B}_{11}(t+1)$ are not changed by the column permutations.

The matrices $\Phi_2(t+1)$ and $\hat{B}_{12}(t+1)$ are updated by formulas

$$\begin{aligned} [\Phi_2(t+1)]' &= \Phi_2(t+1)E_k \quad , \\ [\hat{B}_{12}(t+1)]' &= \hat{B}_{12}(t+1)E_k \quad . \end{aligned} \tag{6.15}$$

As

$$\hat{B}_2^1(t+1) = \hat{B}_{12}(t+1) - \hat{B}_{0B}(t+1)\Phi_2(t+1)$$

then, taking into account (6.15), we obtain

$$[B_2^1(t+1)]' = B_2^1(t+1)E_k \quad . \tag{6.16}$$

Similar reasoning is valid up to the step t^*. Thus, the interchange of the q-th column of the matrix $\hat{D}_{0B}(\tau)$ with k-th column of the matrix $\hat{D}_{0B}(t^*)$ causes changes neither in the local bases $\hat{D}_{0B}(\tau)$ nor in the matrices $\hat{B}_{0B}(\tau)$ $(\tau = t+1,\ldots,t^* - 1)$; the matrices $\Phi_2(\tau)$ and $B_2^1(\tau)$ are updated by formulas (6.15), (6.16) if $t+1 = \tau(\tau = t+1, t+2,\ldots,t^* - 1)$.

At step t^*, part of the columns of the matrix $G(t^*)B_2^1(t^* - 1)$ enters the matrix $\hat{D}_{0B}(t^*)$. Therefore, the updating of the matrices at this step reduces to the case considered above.

This procedure we shall call *the interchange of the ℓ-th column of the matrix $\hat{D}_{0B}(t)$ with the k-th column of the matrix $\hat{D}_{0B}(t^*)$, where $t^* > t + 1$.*

The procedures of column permutation of the matrices $\hat{D}_{0B}(t)$ and $\hat{D}_{0B}(t^*)$ $(t^* > t+1)$ allow us to describe the updating procedure of the old local bases $\{\hat{D}_{0B}(t)\}$ into new ones $\{\hat{D}_{0B}(t)\}'$.

When a vector $\hat{d}_{0\ell}(t_2)$ is replaced by a vector $d_j(t_1)$, two cases are possible.

Case 1: $t_2 < t_1$

In this case, the ℓ-th row of the matrix $\Phi_B(t)$ contains a nonzero pivot element. In fact, the index of the outgoing variable is defined by the relation (3.8). Hence the ℓ-th component of the vector $\hat{v}_{0B}(t_2)$ is not zero.

From (2.8) and (3.7), we find that

$$\hat{v}_{0B}(t_2) = -\Phi_B(t_2)\hat{v}_{1B}(t_2) \text{ if } t_2 < t_1 \quad .$$

Therefore, the ℓ-th row of the matrix $\Phi_B(t_2)$ contains at least one non-zero element.

Let the pivot element correspond to the j-th component of vector $\hat{u}_{0B}(t_2 + \tau)$.

Replace the ℓ-th column of the matrix $\hat{D}_{0B}(t_2)$ by the j-th column of the matrix $\hat{D}_{0B}(t_2 + \tau)$. This interchange does not change the basic solution. Therefore, if $t_2 + \tau < t_1$, the above reasonings are true and we can proceed with the interchanges. In result, we obtain the following case.

Case 2: $t_2 \geq t_1$

Proceeding with these subsequent interchanges, we remove the outgoing vector into such a local basis $\hat{D}_{0B}(t_3)$, $t_3 \geq t_1$, which satisfies the condition of Theorem 6.1.

If such $t_3 \leq T - 1$ does not exist, then we replace the outgoing column into the last local basis $\hat{D}_{0B}(T - 1)$.

In turn, the outgoing column can be replaced in the local basis $\hat{D}_{0B}(t_3)$.

Let the outgoing vector be the ℓ-th column of the matrix $\hat{D}_{0B}(t_3)$. Before introducing the vector $d_j(t_1)$ into the basis, it is necessary to recompute it at the step t_3.

In result we obtain

$$\begin{aligned}
\hat{v}^*_{0B}(t_1) &= \hat{D}^{-1}_{0B}(t_1)d_j(t_1) \quad , \\
y^*(t_1 + 1) &= -b_j(t_1) + \hat{B}_{0B}(t_1)\hat{v}^*_{0B}(t_1) \\
\hat{v}^*_{0B}(\tau) &= -\hat{D}^{-1}_{0B}(\tau)G(\tau)y^*(\tau) \quad , \\
y^*(\tau + 1) &= A(\tau)y^*(\tau) + \hat{B}_{0B}(\tau)\hat{v}^*_{0b}(\tau) \quad , \\
\tau &= t_1 + 1,\ t_1 + 2, \ldots, t_3 \quad .
\end{aligned} \tag{6.17}$$

In these formulas, the new local bases $\{\hat{D}_{0B}(t)\}$ are used.

The above considered updating of the ingoing column $d_j(t_1)$ is possible as the ℓ-th (pivot) element of the vector $\hat{v}^*_{0B}(t_3)$ is not zero.

In fact, the ℓ-th element of the vector $\hat{v}^*_{0B}(t_3)$ is not zero, in accordance with (3.8) and the updating formulas (6.17) coincide with the formulas (3.6) and (3.7).

In accordance with (2.8) and (3.7)

$$\hat{v}_{0B}(t_3) = \hat{v}_{0B}(t_3) - \Phi_B(t_3)\hat{v}_{1B}(t_3) \quad .$$

But as the ℓ-th row of the matrix $\Phi_B(t_3)$ vanishes, $\hat{v}_{0\ell}(t_3) = \hat{v}^*_{0\ell}(t_3) \neq 0$. Thus a new set of local bases is obtained.

7. General Scheme of the Dynamic Simplex Method

Let at some iteration there be known: $\{\hat{D}_{0B}^{-1}(t)\}$, the inverse bases; $\{\hat{u}_{0B}(t)\}$, the basic feasible control; $\{x(t)\}$, the corresponding trajectory; $\{\lambda(t), p(t)\}$, the dual variables (simplex-multipliers).

As in the static simplex method, one can introduce artificial variables at zero iteration if necessary. In that case, the zero iteration local bases are the identity matrices.

In accordance with Sections 3 to 6, the general procedure of the dynamic simplex method comprises the following stages:

1. Choose some pair of indices (j,t_1), for which $\Delta_j(t_1) < 0$, $(j,t_1) \in I_N(u)$, where $\Delta_j(t_1)$ are defined from Section 5. Usually, a pair (j,t_1) with maximal absolute value of $\Delta_j(t_1)$ is selected. If all $\Delta_j(t_1) \geq 0$, $(j,t) \in I_N(u)$, then we have an optimal solution of the problem.

2. Define sequences of vector coefficients $\{v,y\}$ from (3.6) and (3.7).

3. Find the indices (ℓ,t_2) for the outgoing column from (3.8). If all $\hat{v}_{0i}(t) \leq 0$, then, the solution is unbounded.

4. Compute the new basic feasible control $\{u'(t)\}$:

$$u_i'(\tau) = \begin{cases} u_i(\tau) - \theta_0 \hat{v}_{iB}(\tau), & (i,\tau) \in I_B(u) \\ \theta & , (i,\tau) = (j,t_1) \\ 0 & , (i,\tau) \in I_N(u), \ (i,\tau) \neq (j,t_1) \end{cases} .$$

5. Update the local bases:

   a) set $t = t_2$

   b) if $t \geq t_1$, then go to stage e);

   c) select the non-zero element in the pivot row of the matrix $\Phi_B(t)$. (The index of the pivot row equals the index of the outgoing column).

d) let the pivot element of the matrix $\Phi_B(t)$ correspond to the component of the basic control, which was re-computed into the local basis at step $t+\tau$. Then

- interchange the variables between local bases $\hat{D}_{0B}(t)$ and $\hat{D}_{0B}(t+\tau)$
- set $t \to t+\tau$
- go to stage b;

e) if $t = T-1$, then go to stage g;

f) if the pivot element of $\Phi_B(t)$ is nonzero, go to c;

g) replace the column to be removed by the column to be introduced into $\hat{D}_{0B}(t)$.

6. Compute the dual variables $\{\lambda, \dot{p}\}$ from (4.2). Go to stage 1.

It should be noted that only an outline of the algorithm is given here. The concrete implementation of the algorithm depends on the specifics of a problem, the type of computer, the strategy used as to which column selected and introduced into (or removed from) the set of local bases, etc.

8. Degeneracy

It was assumed above that all basic feasible controls were nondegenerate.

This assumption was necessary in order to guarantee that for each successive set of local feasible bases, the associated value of the objective function is larger than those that precede it. This guarantees that we will reach the optimal solution in a finite number of possible sets of local feasible bases.

For the degenerate case, there is the possibility of computing a θ_0 at step 3 of the method, for which $\theta_0 = 0$. Therefore, the selection of a vector to be removed from and a vector to be introduced into the set of local bases will give a new basic feasible control with the value of the objective function being equal to the preceding one. Theoretical examples have been constructed to show that in this case cycling of the procedure is possible. In practical examples this has never happened (with one possible exception). In order to protect against this possibility, a special rule for selecting the outgoing column can be introduced to prevent cycling in the case of degeneracy.

Here we can use the method of overcoming degeneracy of the simplex method [3]. For this we need the columns of the inverse $\bar{B}^{-1}$ (see (2.5)). The j-th column y_j of the inverse $\bar{B}^{-1}$ is a solution of the system of equations:

$$\bar{B}y_j = e_j \quad , \tag{8.1}$$

where e_j is the unit vector of dimension $(m+n)T$ with the j-th component equal to one.

The system (8.1) can be solved by using the factorized representation of the basis matrix, which is similar to the primal solution procedure (Section 3).

9. Numerical Example

Experimental results of tests with the algorithm and its numerical evaluation will be described in a separate paper. Here we consider an illustrative numerical example and give a theoretical evaluation (Section 10) of the method.

We consider the problem with scalar state equations and constraints (that is, $n = m = 1$). In this case, the dimension of the "global" basis matrix will be $2T \times 2T$, hence the corresponding static LP problem is not a very trivial one for large T. Using the dynamic simplex method, we do not need to invert the global basis; what is more, we do not need, for a considered example, to invert local bases either, because if $m = 1$, the local bases are simply numbers.

Problem: Given the state equations

$$x(t+1) = x(t) + u(t) - v(t) \quad (t = 0,\ldots,4) \tag{9.1}$$

with

$$x(0) = 0 \tag{9.2}$$

where $x(t)$, $u(t)$, $v(t)$ are scalars. Find $\{u(0),\ldots,u(4)\}$, $\{v(0),\ldots,v(4)\}$ and $\{x(0),\ldots,x(5)\}$ which satisfy (9.1), (9.2) and constraints

$$x(t) + u(t) + v(t) = f(t) \tag{9.3}$$

$$u(t) \geq 0 \quad ; \quad v(t) \geq 0$$

where $f(0) = 10$; $f(1) = 5$; $f(2) = 5$; $f(3) = 10$; $f(4) = 5$ and minimize

$$J = 10x(5) \quad .$$

The tableau form of the problem is given below

u(0)	v(0)	x(1)	u(1)	v(1)	x(2)	u(2)	v(2)	x(3)	u(3)	v(3)	x(4)	u(4)	v(4)	x(5)	
1	1														= 10
1	−1	−1													= 0
		1	1	1											= 5
		1	1	−1	−1										= 0
					1	1	1								= 5
					1	1	−1	−1							= 0
								1	1	1					= 10
								1	1	−1	−1				= 0
											1	1	1		= 5
											1	1	−1	−1	= 0

Thus, if we solve the problem by the standard simplex method, it is necessary to handle with 10×10 "global" basis at each iteration.

Now we proceed by the dynamic algorithm. Let $\{u^{(0)}(0), v^{(0)}(0), x^{(0)}(1), x^{(0)}(2), u^{(0)}(2), x^{(0)}(3), u^{(0)}(3), v^{(0)}(3), x^{(0)}(4), v^{(0)}(4)\}$ be the first basic variables. The corresponding local bases $\hat{D}_{0B}(t)$ $(t = 0,\ldots,4)$ are the following:

$$\hat{D}_{0B}^{(0)}(0) = 1 \; ; \qquad \hat{u}_{0B}^{(0)}(0) = u(0)$$
$$\hat{D}_{0B}^{(0)}(1) = -2 \; ; \qquad \hat{u}_{0B}^{(0)}(1) = v(0)$$
$$\hat{D}_{0B}^{(0)}(2) = 1 \; ; \qquad \hat{u}_{0B}^{(0)}(2) = u(2)$$
$$\hat{D}_{0B}^{(0)}(3) = 1 \; ; \qquad \hat{u}_{0B}^{(0)}(3) = u(3)$$
$$\hat{D}_{0B}^{(0)}(4) = -2 \; ; \qquad \hat{u}_{0B}^{(0)}(4) = v(3)$$
$$\hat{D}_{0B}^{(0)}(5) = -2 \; ; \qquad \hat{u}_{0B}^{(0)}(5) = v(4) \quad .$$

Note that control variable $v(0)$ from step $t = 0$ enters the local basis $\hat{D}_{0B}^{(0)}(1)$ at the next step $t = 1$. As variable $x(5)$ does not enter the "global" basis on this iteration, it is necessary to introduce an additional local basis $\hat{D}_{0B}^{(0)}(5)$ which corresponds to variable $v(4)$.

The corresponding set of $\Phi_B(t)$ and $B_B^1(t)$ $(t = 0,..,4)$ are the following:

$$\Phi_B^{(0)}(0) = 1 \; ; \qquad B_B^{1(0)}(0) = -2$$
$$\Phi_B^{(0)}(3) = 1 \; ; \qquad B_B^{1(0)}(3) = -2 \qquad (9.6)$$
$$\Phi_B^{(0)}(4) = -0.5 \; ;$$

$\Phi_B^{(0)}(1)$; $\Phi_B^{(0)}(2)$, $B_B^{1(0)}(1)$, $B_B^{1(0)}(2)$, $B_B^{1(0)}(4)$ are zeros.

Using (3.4) and (3.5) for (9.1), (9.2) and (9.5), we obtain the first primal solution:

$$u^{(0)}(0) = 7.5 \qquad u^{(0)}(2) = 0 \quad u^{(0)}(3) = 2.5 \qquad (9.7)$$
$$v^{(0)}(0) = 2.5 \qquad v^{(0)}(3) = 2.5 \quad v^{(0)}(4) = 5$$
$$x^{(0)}(1) = 5 \quad x^{(0)}(2) = 5 \quad x^{(0)}(3) = 5 \quad x^{(0)}(4) = 5$$

the value of objective function: $x^{(0)}(5) = 0$.

As coefficients of the objective function for basic variables are zeros, then all simplex-multipliers (according to (4.2)) are also zeros. Therefore, we have all Δ_j are zeros but $\Delta^{(0)}(x(5)) = -10$. Hence, $x(5)$ is to be introduced to the basis.

Denoting coefficients $\hat{v}_{0B}(t)$ for variables u(t), v(t) and x(t) as $\alpha(t)$, $\beta(t)$ and $\gamma(t)$ respectively, we calculate using (3.6) and (3.7), that $\alpha^{(0)}(3) = -0.25$; $\beta^{(0)}(3) = 0.5$; $\beta^{(0)}(4) = 0.5$; $\gamma^{(0)}(4) = -0.25$ the other $\alpha^{(0)}(t)$, $\beta^{(0)}(t)$ and $\gamma^{(0)}(t)$ are zeros.

From (3.8)

$$\theta_0^{(0)} = \min \left\{ \frac{2.5}{-0.25} ; \frac{2.5}{0.25} ; \frac{5}{0.5} \right\} = -10$$

(it should be taken into account that {x(t)} are free variables). Thus, $x^{(1)}(5) = \theta_0^{(0)} = -10$ and u(3) is to be removed from the basis.

The new primal solution will be the following

$u^{(1)}(0) = 7.5$; $u^{(1)}(2) = 0$

$v^{(1)}(0) = 2.5$; $v^{(1)}(3) = 5$ $v^{(1)}(4) = 10$

$x^{(1)}(1) = 5$; $x^{(1)}(2) = 5$; $x^{(1)}(3) = 5$; $x^{(1)}(4) = 0$ $x^{(1)}(5) = -10$

Now old local bases (9.5) are updated. As variable u(3) leaves the basis, we have to interchange variables u(3) and v(3). After interchange: $\hat{D}_{0B}(3) = 1$, $\Phi_B(3) = 1$; $B_B^1(3) = 2$; $\hat{D}_{0B}(4) = 2$; $\Phi_B(4) = 0.5$.

Then u(3) and v(4) should be interchanged. Hence $\hat{D}_{0B}(4) = 1$; $\Phi_B(4) = 2$; $\hat{D}_{0B}(5) = 4$. Finally, we replace u(3) by x(5), then $\hat{D}_{0B}(5) = -1$.

Thus, the updated local bases are

$$\begin{array}{ll} \hat{D}_{0B}^{(1)}(0) = 1 & \hat{D}_{0B}^{(1)}(3) = 1 \\ \hat{D}_{0B}^{(1)}(1) = -2 & \hat{D}_{0B}^{(1)}(4) = 1 \\ \hat{D}_{0B}^{(1)}(2) = 1 & \hat{D}_{0B}^{(1)}(5) = 4 \quad . \end{array} \tag{9.8}$$

We can begin new iterations now. Using (4.2), the dual solution is obtained for local bases (9.8):

$$
\begin{array}{lll}
p^{(1)}(5) = 10 & p^{(1)}(3) = 40 & p^{(1)}(1) = 0 \\
\lambda^{(1)}(4) = -10 & \lambda^{(1)}(2) = 40 & \lambda^{(1)}(0) = 0 \\
p^{(1)}(4) = 20 & p^{(1)}(2) = 0 & \\
\lambda^{(1)}(3) = -20 & \lambda^{(1)}(1) = 0 &
\end{array}
\tag{9.9}
$$

From (9.9) and (5.1), $\Delta^{(1)}(u(4)) = -20$; $\Delta^{(1)}(u(3)) = -60$; $\Delta^{(1)}(v(2)) = 80$, the other $\Delta^{(1)}$ are zeros. Hence, variable v(2) should be introduced into local bases. Calculating θ_0 for this iteration, we find that $\theta_0^{(1)} = 0$ and u(2) should be removed from the bases. As $\Phi_B^{(1)}(2) = 0$ and variables u(2) and v(2) are from the same step $t = 2$, only local basis $\hat{D}_{0B}(2)$ at this step $t = 2$ must be updated. In result, $\hat{D}_{0B}(2) = 1$ and the other local bases have the same values as in (9.8). The new iteration yields

$$
\begin{array}{lll}
p^{(2)}(5) = 10 & p^{(2)}(3) = 40 & p^{(2)}(1) = 0 \\
\lambda^{(2)}(4) = -10 & \lambda^{(2)}(2) = -40 & \lambda^{(2)}(0) = 0 \\
p^{(2)}(4) = 20 & p^{(2)}(2) = 80 & \\
\lambda^{(2)}(3) = -20 & \lambda^{(2)}(1) = 80 &
\end{array}
\tag{9.10}
$$

and $\Delta^{(2)}(u(4)) = -20$; $\Delta^{(2)}(u(2)) = -80$; $\Delta^{(2)}(v(1)) = 160$; $\Delta^{(2)}(u(3)) = -40$; $\Delta^{(2)}(u(1)) = 0$.

Hence v(1) is introduced to the local bases, $\theta_0^{(2)} = 15$ and u(0) is removed from the local bases. At this iteration, the local bases $\hat{D}_{0B}(0)$ and $\hat{D}_{0B}(1)$ are updated. In result, we obtain

$$
\begin{array}{lll}
v^{(3)}(0) = 10 & v^{(3)}(1) = 15 & v^{(3)}(2) = 30 \\
x^{(3)}(1) = -10 & x^{(3)}(2) = -25 & x^{(3)}(3) = -55 \\
v^{(3)}(3) = 65 & v^{(3)}(4) = 130 & \\
x^{(3)}(4) = -120 & x^{(3)}(5) = -250 &
\end{array}
\tag{9.11}
$$

and $p^{(3)}(2) = 80$; $p^{(3)}(1) = 160$; $\lambda^{(3)}(1) = -80$; $\lambda^{(3)}(0) = -160$, the other $p^{(3)}(t)$ and $\lambda^{(3)}(t)$ are the same as in (9.10). All values of $\Delta^{(3)}(\cdot)$ are negative now. Therefore, (9.11) is an optimal solution.

10. Evaluation of Algorithm

Above we considered an illustrative numerical example which is not so easy to solve by hand using the conventional "static" simplex method, but is very simple to handle by the dynamic algorithm.

Now we give some theoretical evaluation of the dynamic simplex method.

As can be seen from Section 7, for realization of the algorithm it is sufficient to operate only with the matrices $\hat{D}_{0B}^{-1}(t)$; $\Phi_B(t)$, $\hat{B}_{0B}(t)$, $B_B^1(t)$, $G(t)$, $A(t)$ $(t = 0,1,\ldots,T-1)$.

Theorem 10.1: The number of columns of matrices $\Phi_B(t)$ *and* $B_B^1(t)$ *does not exceed* n.

Proof: Let 2T steps of the factorization process be carried out.

Then the formula (2.7) can be rewritten as

$$\bar{B} = \bar{B}_{2t-1}V_{t-1}U_{t-1}\cdots V_0U_0 \quad .$$

On the main diagonal of the matrix $\bar{B}_{2t-1}$ there is the submatrix

$$F = \begin{bmatrix} \hat{D}_{0B}(t) & \hat{D}_{1B}(t) \\ \hat{B}_{0B}(t) & \hat{B}_{1B}(t) \end{bmatrix} \quad .$$

The columns of the submatrix F are linearly independent as the matrix B_{2t-1} is nonsingular. Consequently, the number of columns of matrices $\hat{D}_{1B}(t)$ and $\hat{B}_{1B}(t)$ cannot be larger than n. Hence, one can obtain the statement of the theorem.

The matrices $\hat{D}_{0B}^{-1}(t)$, $\hat{B}_{0B}(t)$, $G(t)$, $A(t)$ have dimensions $(m\times m)$, $(n\times m)$, $(m\times n)$, $(n\times n)$ respectively. Therefore, the algorithm operates only with the set of T matrices, each containing no more than m or n columns.

At the same time, the straightforward application of the simplex method to Problem 1.1 (in the space of $\{u,x\}$) leads to the necessity of operating with the basis matrix of dimension $(m+n)T \times (m+n)T$ or of dimension $mT \times mT$, if the state variables are excluded beforehand.

Thus, in some respects, the dynamic simplex method realizes a decomposition of the problem that allows a substantial saving in the number of arithmetical operations and in the core memory.

As was mentioned above, the DLP Problem 1.1 can be considered as some "large" static LP problem and thus the simplex method can be used for its solution. Let us find an upper estimation of a number of iterations. At each iteration, the simplex method requires no more than k^2 multiplications for updating of the inverse, where k is the number of rows of the basic matrix. Hence, the total number of multiplications for the basis updating is no more than $(m+n)^2T^2$. To compute the coefficients which express the column to be introduced into a basis in terms of columns of the current basis, the simplex method requires some $(m+n)^2T$ multiplications.

Now we shall evaluate the number of multiplications for the dynamic simplex method. It was shown that at one interchange, the local bases are updated by multiplication on the elementary column or row matrix. The interchange of columns between two neighboring local bases $\hat{D}_{0B}(t)$ and $\hat{D}_{0B}(t+1)$ requires no more than $3(m+n)^2$ multiplications. (The matrices $\hat{D}_{0B}^{-1}(t)$, $\hat{B}_{0B}(t)$, $\Phi_B(t)$, $B_B^1(t)$, $\hat{D}_{0B}^{-1}(t+1)$, $\hat{B}_{0B}(t+1)$, $\Phi_B(t+1)$ are updated). In the worst case, when the outgoing column from the local bases $\hat{D}_{0B}(0)$ is entered into the local basis $\hat{D}_{0B}(T-1)$, one needs T interchanges. We assume that the average number of interchanges is T/2. Thus the dynamic simplex method requires approximately $1.5(m+n)^2T$ multiplications for local bases updating per iteration.

Calculation of the coefficients expressing the ingoing vector requires about $(m+n)^2T$ multiplications. In addition, local bases can be represented in factorized form, thus enabling use of the effective procedures of static LP [3].

Solution of Problem 1.1 by the static simplex method requires storage of the inverse of dimension $(m+n)T \times (m+n)T$. The dynamic simplex method requires storage of only T matrices of dimension $m \times m$ $(\hat{D}_{0B}^{-1}(t), \hat{B}_{0B}(t))$ and $T-1$ matrices of dimension $m \times n$ $(\Phi_B(t))$ and $n \times n$ $(B_B^1(t))$.

Thus, comparing the estimates of the static and dynamic algorithms for solution of Problem 1.1, one can see that the volume of computation and the core memory increases linearly with T for the dynamic algorithm and by quadratic law for the static algorithm.

It is more important that only part of the local bases be updated at each iteration. Therefore the dynamic simplex method may turn out to be superior in comparison with a conventional revised simplex algorithm not only because it offers a more compact substitute for the basic inverse but also because it allows the use of only a part of the basic inverse representation required at each iteration.

11. Dual Algorithms

The introduction of local bases and techniques of their handling allows us to develop dual and primal-dual versions of the dynamic simplex method. The main advantage of using the dual methods is that the dual statements of many problems have explicit solutions. The other is connected with the choice of different selection strategies to the vector pair which enters and leaves the basis.

In the primal version of the dynamic simplex method, there are some options for choice of a column with the most negative price from all non-basic columns or from some set of these columns, etc. But a column to be removed from the basis is unique in the nondegenerate case.

Contrarily, in dual methods, there are options in the choice of a column to be removed from the basis. It can be effectively used in dual versions of the method. In practical problems, local bases $\{\hat{D}_{0B}(t)\}$ can be rather large, therefore part of the local bases should be stored at the external storage capacities. Input-output operations are comparatively time-consuming. Hence, to reduce the total solution time, it is desirable to have more pivoting operations with a given local basis.

Thus, the usage of different dual and primal-dual strategies allows us to adjust the algorithm to the specifics of the computer to be used and to the problem to be solved.

12. Extensions

The approach considered above is flexible and allows different extensions and generalizations. Below, we describe briefly two of them.

First, in Problem 1.1, the state variables $x(t)$ are considered to be free. The case when $x(t) \geq 0$ or $0 \leq x(t) \leq \alpha(t)$ can be treated by the approach very easily. In fact, from the point of view of the computer implementation of the algorithm, it is better to handle with the multiplicative form of the inverse of

$$\tilde{D}_{0B}(t) = \begin{pmatrix} \hat{D}_{0B}(t) & 0 \\ \hat{B}_{0B}(t) & -I \end{pmatrix}$$

rather than with $\hat{D}_{0B}^{-1}(t)$, because the addition of the unit matrix $-I$ does not generate additional zeros in the "eta-file". If $x(t)$ are not constrained, then by handling with the inverse of $\tilde{D}_{0B}(t)$ we can consider the rows corresponding to low blocks of $\tilde{D}_{0B}(t)$, that is, $\hat{B}_{0B}(t)$ and $-I$, as free. In this case, all $x(t)$ are in the basis.

If $x(t) \geq 0$, then the state variables $x(t)$ should be handled in the same way as control variables $u(t) \geq 0$. In this case, not all $x(t)$ will be in the basis.

Evidently, this includes the case when both state and control variables have upper bound constraints. (The inclusion of generalized upper bound constraints is also possible).

The second case, which has many important applications, is DLP with time delays. Instead of (1.1) and (1.3), we have in this case

$$\begin{aligned} x(t+1) &= \sum_{\nu} A(t,\tau_{\nu})x(t-\tau_{\nu}) + \sum_{\mu} B(t,\tau_{\mu})u(t-\tau_{\mu}) \\ &\sum_{\nu} G(t,\tau_{\nu})x(t-\tau_{\nu}) + \sum_{\mu} D(t,\tau_{\mu})u(t-\tau_{\mu}) \qquad (12.1) \\ &= f(t) \end{aligned}$$

with given values for $x(t)$ and $u(t-1)$ if $t \leq 0$. Here $\{\tau_{\nu}\}$, $\{\tau_{\mu}\}$ are given ordered sets of integers.

New submatrices will appear to the left from the main staircase of the diagonal of B* in (2.7a) (see Figure 1a and b).

```
X X                     X X                     X X
X X X                   X X X                   X X X
    X X X                   X X X               X X X X X
    X X X X                 X X X X             X X X X X X
          X X X             X     X X X               X X X X X
          X X X X           X     X X X X             X X X X X X
                X X X             X     X X X               X X X X X
                X X X X           X     X X X X             X X X X X X

        a                         b                         c
```

Figure 1.

Because the main staircase structure is not changed in this case (Figure 1), we can use the same procedure as in the case without time delays. There will be only one difference. Now local bases $\hat{D}_{0B}(t)$ will contain recomputed columns both from previous steps $\tau < t$ and columns from time "delayed" matrices $D(t,\tau)$ $\tau \leq t$, which enter the constraints (12.1) at step t.

Thus, both of these important extensions of Problem 1.1 can be handled by the algorithm almost without any modifications.

The extensions considered above concern the extension of Problem 1.1 within the DLP framework. It should be underlined that the approach is also applicable to solve LP problems with general structure (such as in Figure 1, if by χ one means some arbitrary matrix).

In this case, the approach will be related to factorization methods considered in [4,5].

13. Conclusion

The general scheme and basic theoretical properties of the dynamic simplex method specially developed for solution of dynamic linear programs have been described and discussed.

Theoretical reasonings show that this algorithm may serve as a base for developing effective computer codes for the solution of DLP problems. However, the final judgment of the efficiency of the algorithm can be made only after a definite period of its exploitation in practice.

It should also be very interesting to compare (both from the theoretical and the computational point of view) the approach given in this paper with the finite-step DLP algorithms based on the Dantzig-Wolfe decomposition principle [6,7,8] and other methods of solving structured LP problems [4-9].

References

[1] Krivonozhko, V.E. and A.I. Propoi, *II. The Dynamic Simplex Method: General Approach*, this issue.

[2] Gantmacher, F.R., *The Theory of Matrices*, Chelsea Publishing Company, New York, 1960.

[3] Dantzig, G.B., *Linear Programming and Extensions*, University Press, Princeton, N.J., 1963.

[4] Bulavskii, V.A., R.A. Zvjagina and M.A. Yakovleva, Numerical Methods of Linear Programming (Special Problems) *Nauka*, 1977, (in Russian).

[5] Kallio, M. and E. Porteus, Triangular Factorization and Generalized Upper Bounding Techniques, *Operations Research*, 25, 1 (1977).

[6] Ho, J.K. and A.S. Manne, Nested Decomposition for Dynamic Models, *Mathematical Programming*, 6, 2 (1974).

[7] Krivonozkho, V.E., Decomposition Algorithm for Dynamic Linear Programming Problems, Izvestia AN SSSR, *Technicheskya Kibernetika*, 3, (1977), 18-25, (in Russian).

[8] Beer, K., Lösung großer linearer Optimierungsaufgaben, VEB Deutscher Verlag der Wissenschaften, Berlin, 1977, (in German).

[9] Wollmer, R.D., A Substitute Inverse for the Basis of a Staircase Structure Linear Program, *Math. of Operations Research*, 2, 3 (1977).

PAPERS IN DYNAMIC LINEAR PROGRAMMING AND RELATED TOPICS

Models

[1] Haefele, W. and A.S. Manne, *Strategies for a Transition from Fossil to Nuclear Fuels*, RR-74-7, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1974.

[2] Suzuki, A., *An Extension of the Haefele-Manne Model for Assessing Strategies for a Transition from Fossil Fuel to Nuclear and Solar Alternatives*, RR-75-047, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1975.

[3] Dantzig, G.B., A Pilot Linear Programming Model for Assessing Physical Impact on the Economy of a Changing Energy Picture, IIASA Conference '76, May 1976.

[4] Propoi, A.I., *Problems of Dynamic Linear Programming*, RM-76-78, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1976.

[5] Haefele, W. and A. Makarov, Modeling of Medium and Long-Range Energy Strategies, Discussion paper given at the workshop on Energy Strategies, International Institute for Applied Systems Analysis, May 1977.

[6] Agnew, M., L. Schrattenholzer and A. Voss, *A Model for Energy Supply Systems and their General Environmental Impact*, RM-78-26, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1978.

[7] Grenon, M. and I. Zimin, Resources Model, Discussion paper given at the workshop on Energy Strategies, International Institute for Applied Systems Analysis, May 1977.

[8] Zimin, I., Aggregated Dynamic Model of a National Economy, Discussion paper given at the workshop on Energy Strategies, International Institute for Applied Systems Analysis, May 1977.

[9] Propoi, A. and F. Willekens, *A Dynamic Linear Programming Approach to National Settlement System Planning*, RM-77-8, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1977.

[10] Propoi, A., *Dynamic Linear Programming Models for Live-Stock Farms*, RM-77-29, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1977.

[11] Csåki, C., *Dynamic Linear Programming Model for Agricultural Investment and Resources Utilization Policies*, RM-77-36, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1977.

[12] Carter, H., C. Csåki and A. Propoi, *Planning Long Range Agricultural Investment Projects: A Dynamic Programming Approach*, RM-77-38, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1977.

[13] Propoi, A., Dynamic Linear Programming Models in Health Care Systems, in *Modeling Health Care Systems*, eds. R. Gibbs and E. Shigan, CP-77-8, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1977.

[14] Propoi, A., *Educational and Manpower Models: A Dynamic Linear Programming Approach*, RM-78-20, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1978.

Methods

[1] Winkler, C., *Basis Factorization for Block Angular Linear Programs: Unified Theory of Partitioning and Decomposition Using the Simplex Method*, RR-74-22, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1974.

[2] Orchard-Hays, W.M., *Some Additional Views on the Simplex Method and the Geometry of Constraint Space*, RR-76-3, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1974.

[3] Propoi, A., *Problems of Dynamic Linear Programming*, RM-76-78, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1976.

[4] Propoi, A., *Dual Systems of Dynamic Linear Programs*, RR-77-9, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1977.

[5] Propoi, A. and V. Krivonozhko, *The Dynamic Simplex Method*, RM-77-24, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1977.

[6] Agnew, M., L. Schrattenholzer and A. Voss, *User's Guide for the MESSAGE Computer Program*, RM-78-26, Addendum, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1978.

[7] Propoi, A. and V. Krivonoshko, *Simplex Method for Dynamic Linear Programs*, RR-78- , International Institute for Applied Systems Analysis, Laxenburg, Austria, 1978.

RELATED IIASA PUBLICATIONS

Basis Factorization for Block-Angular Linear Problems: Unified Theory of Partitioning and Decomposition using the Simplex Method. C. Winkler. (RR-74-022) $3.00 AS45. MICROFICHE ONLY.

Some Additional Views on the Simplex Method and the Geometry of Constraint Space. Wm. Orchard-Hays. (RR-76-003) $4.50 AS80.

Dual Systems of Dynamic Linear Programming. A.I. Propoi. (RR-77-009) $1.50 AS30.

Problems of Dynamic Linear Programming. A.I. Propoi. (RM-76-078) $3.00 AS45. (MICROFICHE ONLY).

A Dynamic Linear Programming Approach to National Settlement System Planning. A.I. Propoi, F. Willekens. (RM-77-008) $3.00 AS45. (MICROFICHE ONLY).

On the Theory of Max-Min. A.I. Propoi. (RM-77-013) $1.50 AS30.

The Dynamic Simplex-Method. A.I. Propoi, V.E. Krivonozhko. (RM-77-024) $3.50 AS60.

Dynamic Linear Programming Models for Livestock Farms. A.I. Propoi. (RM-77-029) $1.50 AS30.

Planning Long Range Agricultural Investment Projects: A Dynamic Linear Programming Approach. H. Carter, C. Csåki, A.I. Propoi. (RM-77-038) $1.50 AS30.

Models for Educational and Manpower Planning: A Dynamic Linear Programming Approach. A. Propoi. (RM-78-020) $6.00 AS85.

Please cite publication number when making an order. See inside back cover for order information.