Journal of the Operations Research
Society of Japan
Vol.21, No.2, June, 1978

A TWO-LEVEL ALGORITHM FOR TWO-STAGE LINEAR PROGRAMS

Tatsuo Aonuma
Kobe University of Commerce

(Received May 25, 1976; Final October 1, 1977)

Abstract An algorithm for solving 2-stage linear programs, especially useful for a program arising from a nested multi-level approach to multi-period planning [3] is presented. The proposed algorithm is of a two-level scheme and similar to Beale's method [4], but the procedure is more systematic and simpler. Computational experience is reported for a series of small test problems. The results show that our algorithm is more efficient than a Beale-like method, and moreover that the number of times for adjusting given initial values of linking variables is unexpectedly small for the test problems, which may also imply an aspect of usefulness of the algorithm for large real problems, especially when a good initial value can be obtained for the linking variable. The algorithm can be also extended to a weakly coupled three or more-stage case.

1. Introduction.

The following dynamic linear program is very familiar to us in real planning problems, especially, arising in multi-period planning such as production planning, economic planning and so on :

$$\begin{aligned} &\text{Maximize} && \sum_{i=1}^{N} c^i x^i \\ &\text{s.t.} \\ &&& A_0^i x^i && = b_0^i \\ &&& A^i x^i + I y^i - I y^{i-1} && = b^i \\ &&& x^i, y^i \geqq 0 \quad (i=1,2,\ ,\ ,\ N), && y^0 = y^N = 0, \end{aligned} \tag{1.1}$$

where A_0^i and A^i are $m_i \times n$ and $m \times n$ matrices respectively, c^i is an n-rowvector, and b_0^i, b^i, x^i and y^i are m_i-, m-, n- and m-column vectors respectively. The linking variable y^i denotes an allocation of resources common to two periods.

A part of this paper was presented at the 5th Management Science Colloquium, August 1976, Osaka, Japan.

This type of program generally becomes very large in size, and therefore computational difficulties often arise in solving it by a direct simplex method within the limited capacity of the computing system concerned. On the other hand, we sometimes have question concerning a necessity of solving accurately such a large model in order to prepare a plan for the uncertain future. Because, in real problems, data for the more distant future in the model usually include much uncertainty and it has been often observed that, under fluctuating circumstances, more discrepancy arises between the program for each period obtained by a dynamic model and the achievement of the program, according as it is a program for a more distant future from the planning point. Taking account of this respect, we have presented a simplified approach, called a nested multi-level approach, to multi-period planning in [1] and [3]. In that approach the problem is reformulated by a 2-stage linear program which consists of both the first submodel representing a program for the first period and the second representing a macro-plan for the remaining periods. The latter may be regarded as an aggregated plan for implementing a global grasp of management activities required in the future. The programs for the planning horizon of N periods are approximately obtained by solving a series of those 2-stage programs generated successively.

We present a two-level algorithm useful for solving a 2-stage linear program, especially in the nested multi-level approach. In the nested multi-level approach, we need to deal with two qualitatively distinct submodels, which are connected to each other by linking variables. For that purpose, our algorithm is designed so as to have the following functions:

(1) Two submodels can be separately optimized in a sense of two-level planning [10] [13]. This implies that each submodel can be respectively dealt with at its pertinent organizational unit through a so-called in-house computer network, and, computationally, within a high-speed memory of given size, it can solve larger problems using auxiliary memory than can conventional methods.

(2) The interactive method [7] can be easily dealt with in the algorithm to structure the preference attitude of a decision-maker for each pair of subobjective values of two submodels in the optimizing process.

It is not our real intention that, only from an aspect of computational efficiency, we compare our algorithm with a direct simplex approach in which an expensive computer with large core memory is needed. We would like to consider both the aspect of physical limitations in our own computing system and that of organizational and procedural convenience for using the approach concerned. Algorithms for solving linear programs with the staircase structure have been developed by several authors, for example, [4], [8], [11] and so on.

It seems that the most typical and latest algorithms accompanied by computational experience are Glassey [8] and Ho & Manne [11]. However, those algorithms for a 2-stage case, which are completely equivalent to Dantzig & Wolfe's method [5], are of a column-generation scheme. Accordingly, two subproblems are not separately optimized for given y and a priori information for the initial value of the linking variable can not be completely used. These are the main reasons why the methods are not so suitable for our purpose mentioned above.

The essential idea of our algorithm is similar to Beale's method [4] in respect of both preserving the special structure of the problem and parametrizing the linking variables, but there are some distinctions in its procedure and approach[1)]. In our algorithm, a direction-finding problem and coupling problem will be newly introduced in order to define a plural number of new parameters at every step and to obtain a linear relationship (called Parameter Transformation Matrix later) between the linking variables and the new parameters more systematically. If we restrict the candidates of the entering basic variable to only one variable at each step in our direction-finding problem, then our algorithm is conceptually close to Beale's method except that several steps for defining the new parameters in his algorithm is reduced to solving our coupling problem. In order to investigate the computational efficiency we compared our algorithm with a simple version of ours as restricted to one candidate of the entering basis in the direction-finding problem instead of a direct representation of Beale's method [4], because his procedure did not seem to be quite systematic and suitable for computer programming. The computational experiments show that our algorithm mostly works better than his method. Concerning a comparison with a direct simplex method, a remarkable result from our computational experience was that the number of times of solving the coupling problem, which is required to adjust a given initial value of the linking variable toward the optimum, was considerably less than our estimation for our small test problems. This fact implies an aspect of usefulness of our two-level approach, because, in real problems, mostly we can understand the problem itself well enough to estimate the tendency of the value of the linking variable to attain the optimum and , therefore, may choose a good initial value for the variable.

The algorithm is of a feasible method for a given initial feasible solution and terminates in a finite number of steps. In terms of coordination it involves two aspects of the resource-adjusting and price-adjusting coordinations [6],[10]

---

1) In Beale's algorithm, after one basic variable in the submodel is exchanged, a series of steps follows it, defining one new parameter at an individual step, and that, at some steps among them an additional constraint needs to be added to m constraints for the definitions of the linking variables, i.e., the number of constraints becomes fluctuant. Computational experience for his method has not been seen as yet.

in a two-level system. By making slight modifications, the algorithm also can be effectively applied to weakly coupled three or more-stage cases such that the total number of linking variables is, roughly speaking, less than or equal to the maximum number of constraints among the submodels.

The two-level algorithm will be presented in Section 2 and also the finiteness of the algorithm will be proved with several theorems. In Section 3 computational experience will be reported for a series of small test problems. In the final section an interactive approach to a preference optimization will be described briefly.

2. A Two-level Algorithm.

We attempt to solve the following two-stage problem [P] by a two-level approach[2].

$$
[P]: \quad \begin{array}{lll} \text{Maximize} & c^1x^1 + c^2x^2 & \\ \text{s.t.} & & \\ & A^1x^1 + y = b^1 & \\ & -y + A^2x^2 = b^2 & \\ & x^1, x^2, y \geqq 0 . & \end{array}
$$

Although we assume for the sake of simplicity that the coefficient matrix of y is an identity matrix, it need not be so in our algorithm. A case in which y has nonzero objective coefficient c^0 can be reduced to the above, for example, by replacing either c^1 to $c^1 - c^0A^1$ or c^2 to $c^2 + c^0A^2$. We can also assume in order to simplify the notations without loss of generality that constraints corresponding to $A_0^i x^i = b_0^i$ in (1.1) are void and two submodels have the same number of constraints, as will be easily seen in the future discussion of the algorithm.

First, we decompose [P] into two subproblems P_1 and P_2. For this purpose, let the allocation of common resources be fixed at $y=y^{(k)}$ (k=0,1,2, , ,), so as to make the subproblems feasible. We assume that such an initial value $y^{(0)}$ can be found. Then, [P] can be equivalently represented as follows[3]:

To find x^1, x^2 and λ such that

$$
[P_\lambda]: \quad \begin{array}{lll} Z(y^{(k)}) = \max & c^1x^1 + c^2x^2 & \text{dual variables} \\ \text{s.t.} & & \\ & -T_k\lambda \leqq y^{(k)} & : \ u^0 \\ & A^1x^1 + T_k\lambda = b^1 - y^{(k)} & : \ u^1 \end{array}
$$

2) In general, the objective function may be represented by $U(z^1,z^2)$, where $z^i = c^ix^i$ and U is a concave function and increasing in each z^i. Our algorithm can be easily extended to this type of nonlinear case, as will be mentioned later.
3) A notation x' denotes the transposition of x in this article.

$$- T_k\lambda + A^2x^2 = b^2 + y^{(k)} \quad : \quad u^2$$
$$x^1, x^2 \geqq 0,$$

where λ denotes a parameter needed to adjust optimally the given allocation $y^{(k)}$, and T_k a non-singular m by m matrix which is called Parameter Transformation Matrix (PTM hereafter) to be defined later, and $T_0 = I$ (identity). We assume that P_λ is bounded and that two subproblems generated by $\lambda = 0$ are also bounded for any feasible $y^{(k)}$.

After solving two subproblems $P_1(y^{(k)})$, $Z^1(y^{(k)}) = \max c^1x^1$ s.t. $A^1x^1 = b^1 - y^{(k)}$ and $x^1 \geqq 0$, and $P_2(y^{(k)})$, $Z^2(y^{(k)}) = \max c^2x^2$ s.t. $A^2x^2 = b^2 + y^{(k)}$ and $x^2 \geqq 0$, let B_i be an optimal basis matrix of P_i, x^{i*} the optimal solution to P_i, $S^i(y^{(k)})$ the value of the corresponding basic variable, and π^i the dual solution. Let us define the following coupling problem :

Primal Coupling Problem.

$$Z(\lambda) = \max (\pi^2 - \pi^1)'T_k\lambda \qquad \text{dual variables}$$
$$\text{s.t.}$$
$$P_c(\lambda): \quad - T_k\lambda \leqq y^{(k)} \quad : \quad \mu_0$$
$$B_1^{-1}T_k\lambda \leqq S^1(y^{(k)}) \quad : \quad \mu_1$$
$$- B_2^{-1}T_k\lambda \leqq S^2(y^{(k)}) \quad : \quad \mu_2$$

The dual coupling problem to $P_c(\lambda)$ can be handled more easily since the variables μ_i (i=0,1,2) are nonnegative. Let $D_c(\lambda)$ denotes it and an optimal solution to $D_c(\lambda)$ be $\rho = D_k^{-1}T_k'(\pi^2 - \pi^1) \geqq 0$, where D_k is the basis matrix of $D_c(\lambda)$. The following results are straightforward from the fundamental theorems on linear programming.

Lemma 1. A triplet $(x^{1*}, x^{2*}, y^{(k)})$ is an optimal solution to [P] if and only if there is a dual solution π^i to $P_i(y^{(k)})$ such that i) $\pi^1 \geqq \pi^2$ and $(\pi^1 - \pi^2)'y^{(k)} = 0$, ii) $(\pi^{i\prime}A^i - c^i)x^{i*} = 0$ (i=1,2). And, the corresponding dual solution $(u^{i*}, i=0,1,2)$ to P_λ can be obtained as $u^{0*} = \pi^1 - \pi^2$ and $u^{i*} = \pi^i$ (i=1,2).

Lemma 2. (1) If $(\pi^2 - \pi^1)'T_k = 0$ (i.e., $\rho = 0$), the corresponding subproblem solutions and $y^{(k)}$ constitute an optimal solution to [P]. (2) Let λ^* be an optimal basic solution to $P_c(\lambda)$. Then, $Z(y^*) = Z(y^{(k)}) + \Delta Z(\lambda^*)$ for $y^* = y^{(k)} + T_k\lambda^*$, and $S^1(y^*) = S^1(y^{(k)}) - B_1^{-1}T_k\lambda^*$ and $S^2(y^*) = S^2(y^{(k)}) + B_2^{-1}T_k\lambda^*$ hold. $S^i(y^*)$ corresponds to the simplex criterion of μ^i in $D_c(\lambda^*)$ and y^* that of μ^0. (3) There are at least m zero elements among the components of $S^1(y^*)$, $S^2(y^*)$ and y^* in total. The number of zero elements is exactly m under the non-degeneracy assumption, which we shall assume hereafter.

Let the sets of indices Γ_i (i=0,1,2) be $\Gamma_0 = [j: y_j^* = 0]$ and $\Gamma_i = [$ all

rows such that the corresponding element of $S^i(y^*)$ is zero] (i=1,2). Then, $|\Gamma_0 \smile \Gamma_1 \smile \Gamma_2|$, the number of elements of $\Gamma_0 \smile \Gamma_1 \smile \Gamma_2$, is equal to $m = m_0 + m_1 + m_2$ under the assumption, where $m_i = |\Gamma_i|$. In addition, it is easily noticed that Γ_i corresponds to the basic variables of $D_c(\lambda)$. Let us define a new PTM, T_k^*, and a corresponding parameter λ by

$$T_k^* = T_k(D_k^{-1})' , \tag{2.1}$$

and

$$y = y^* + T_k^* \lambda . \tag{2.2}$$

Then, a canonical form of P_λ for the newly defined parameter λ can be represented in a simplex tableau form as follows :

$$\begin{aligned}
&\text{Maximize} \quad z \\
&\text{s.t.} \\
Z(y^*) &= z + \hat{c}^1 x_N^1 - \rho'\lambda + \hat{c}^2 x_N^2 \\
y^* &= y - T_k^* \lambda \\
S^1(y^*) &= x_B^1 + \hat{A}^1 x_N^1 + B_1^{-1} T_k^* \lambda \\
S^2(y^*) &= -B_2^{-1} T_k^* \lambda + x_B^2 + \hat{A}^2 x_N^2 \\
&y,\ x_B^i,\ x_N^i \geqq 0 \quad (i=1,2),
\end{aligned} \tag{2.3}$$

where x_B^i and x_N^i denote the basic and non-basic variable of each P_i respectively, and $\hat{c}^i \geqq 0$ and $\rho \geqq 0$ [4]. The next lemma follows clearly from the derivation of the primal coupling problem.

Lemma 3. Let $Z(y)$ represent the sum of two subobjective functions $Z^1(y) + Z^2(y)$ for given y. Then, the triplet $(S^1(y^*), S^2(y^*), y^*)$, which is obtained in (2) of Lemma 2, gives the maximum of $Z(y)$ for the optimal bases B_i's chosen in $P_i(y^{(k)})$ (i=1,2).

If we assume for the sake of simplicity that all rows in Γ_i are put in a consecutive order and the first m_0 elements of μ_0 are in Γ_0, the last m_1 elements of μ_1 in Γ_1, and the first m_2 elements of μ_2 in Γ_2, then T^* and the two linking matrices $B_i^{-1}T^*$ in (2.3) have the following structures respectively :

$$T^* = -\begin{bmatrix} I_{m_0} & 0 & 0 \\ U_0 & V_0 & W_0 \end{bmatrix}, \quad B_1^{-1}T^* = \begin{bmatrix} U_1 & V_1 & W_1 \\ 0 & I_{m_1} & 0 \end{bmatrix}, \quad -B_2^{-1}T^* = \begin{bmatrix} 0 & 0 & I_{m_2} \\ U_2 & V_2 & W_2 \end{bmatrix}, \tag{2.4}$$

where U_i is an $(m-m_i)$ by m_0 matrix, V_i an $(m-m_i)$ by m_1 matrix, W_i an $(m-m_i)$ by m_2 matrix. Let us introduce the following notations : λ^i, a part of λ corresponding to I_{m_i} ; ρ^i, the corresponding part of ρ ; $\hat{A}_\Gamma^i$, a part of $\hat{A}^i$ correspond-

4) We may sometimes omit the superfixes k and i hereafter if no confusion arises.

ing to Γ_i. Then, ρ can be written as $\rho' = (\rho^{0\prime}, \rho^{1\prime}, \rho^{2\prime})$.

When we solve two subproblems separately after fixing y at y*, in constrast with Lemma 1 (i) the corresponding shadow price π^1 of the common resource is not necessarily identical to π^2 for y* > 0 even if y* has been optimal to [P]. It is rather more usual by our computational experience that they are different each other. A simple example has been shown in [2], in which the optimal solutions to two subproblems and y* attain an optimal solution to [P] although the corresponding shadow prices π^i's (i=1,2) do not satisfy the relation $(\pi^1 - \pi^2)'y^* = 0$ in Lemma 1. The next theorem states a relationship between the shadow prices of subproblems when each subproblem is separately solved, and the global optimality of [P].

Theorem 1. Let π^i be an optimal dual solution to $P_i(y^*)$. If

$$\hat{u}^{1\prime} = \pi^{1\prime} + (0, \rho^{1\prime})(B_1^{-1}), \quad \text{and} \quad \hat{u}^{2\prime} = \pi^{2\prime} + (\rho^{2\prime}, 0)(B_2^{-1}) \tag{2.5}$$

are dual feasible solutions to P_i(i=1,2) respectively, the triplet (x^{1*}, x^{2*}, y^*) is an optimal solution to [P].

Proof: It is sufficient to show that (i) and (ii) in Lemma 1 hold for $\hat{u}^i$ in (2.5). First, we shall show that (i) holds. Let D be an optimal basis matrix of $D_c(\lambda)$. Then we have $T = (D^{-1})'$ and $\rho' = (\pi^1 - \pi^2)'T$. From (2.4),

$$(\hat{u}^1 - \hat{u}^2)'T = (\pi^1 - \pi^2)'T + (0, \rho^{1\prime})B_1^{-1}T - (\rho^{2\prime}, 0)B_2^{-1}T$$

$$= -\rho' + (0, \rho^{1\prime}, 0) + (0, 0, \rho^{2\prime}) = -(\rho^{0\prime}, 0, 0).$$

This means $(\hat{u}^1 - \hat{u}^2) \geqq 0$ and $(\hat{u}^1 - \hat{u}^2)y^* = 0$, because the first m_0 rows in $D' = T^{-1}$ are equal to $(-I_{m_0}, 0)$. Next,

$$(\hat{u}^{1\prime}A^1 - c^1)x^{1*} = (\pi^{1\prime}A^1 - c^1)x^{1*} + (0, \rho^{1\prime})B_1^{-1}A^1x^{1*}.$$

The first term is zero because π^1 is an optimal solution to $P_1(y^*)$. The second term also becomes zero because all components of $B_1^{-1}A^1x^{1*} = S^1(y^*)$ corresponding to ρ^1 belong to Γ_1 and should be zero from (3) in Lemma 2. This shows that (ii) holds for $\hat{u}^1$. Similarly, we can prove it for $\hat{u}^2$.

In order to improve a feasible solution in the canonical form (2.3) a direction-finding problem should be considered. The direction-finding problem in the primal-dual method is described in a dual form as follows [14]:

Associated Restricted Dual Problem (i = 1,2).

$$\begin{aligned} &\text{Max} \quad -\hat{c}^i x_N^i + \rho^{i\prime}\lambda^i \\ &\text{s.t.} \\ &\quad \hat{A}_\Gamma^i x_N^i + I_{m_i}\lambda^i \leqq 0 \\ &\quad x_N^i \geqq 0, \qquad \lambda^{i\prime} \leqq (K, K, \ , \ , K), \end{aligned} \tag{2.6}$$

where K is any positive number. Let an optimal solution to (2.6) be (x_N^*, λ^*). Then, $-\hat{c}x_N^* + \rho'\lambda^* \geqq 0$ holds because $(x_N = 0, \lambda = 0)$ becomes a feasible solution,

and it is known that a feasible basic solution to (2.3) is optimal if and only if $\max\ (-\hat{c}^i x_N^i + \rho^{i\prime}\lambda^i) = 0$ (i=1,2). An optimal solution (x_N^*, λ^*) to (2.6) with $(-\hat{c}x_N^* + \rho'\lambda^*) > 0$ gives a usable feasible direction to P_λ. There must be at least one positive λ_j^* in the direction because of $\hat{c} \geqq 0$ and $\rho \geqq 0$. Thus, we have the following lemma.

Lemma 4. If P_λ is not optimal, there is a usable feasible direction that has at least one $\lambda_j^* > 0$ among the components.

It is easily seen that $\hat{A}_\Gamma x_N^* + \lambda_\Gamma^* = 0$ holds except a degenerate case because of $\rho \geqq 0$ and the special structure of the constraint matrix of λ in (2.6). Therefore, (2.6) can be rewritten equivalently as $\max\ -(\hat{c} + \rho'\hat{A}_\Gamma)x_N$ subject to $x_N \geqq 0$. If $(\hat{c} + \rho'\hat{A}_\Gamma) \geqq 0$, we have always $-(\hat{c} + \rho'\hat{A}_\Gamma)x_N^* = 0$ and vice versa. Thus, we have proved the following Beale's result by another approach.

Theorem 2 (Beale [4]). Let x^{i*} be an optimal solution to $P_i(y^*)$. Then, the triplet (x^{1*}, x^{2*}, y^*) is an optimal solution to [P] if and only if

(2.7) (i) $\hat{c}^1 + (0, \rho^{1\prime})\hat{A}^1 \geqq 0$, and (ii) $\hat{c}^2 + (\rho^{2\prime}, 0)\hat{A}^2 \geqq 0$.

Considering the result of Lemma 4, we shall attempt to solve the following direction-finding problem restricted to $\lambda^i \geqq 0$, instead of (2.6):

$[F_i]$ (i=1,2): Maximize $p^i x_N^i$ s.t. $\hat{A}_\Gamma^i x_N^i \leqq 0$, $x_N^i \geqq 0$,

where $p^i = -(\hat{c}^i + \rho^{i\prime}\hat{A}_\Gamma^i)$. If (2.3) is not optimal, there is at least one $p_j^i > 0$ from Theorem 2. Each left-hand side of the inequalities (2.7) represents the simplex criterion of x_N after each objective coefficient of the basic variable $x_j \in \Gamma$ is changed from c_j to $c_j + \rho_j$ respectively. In a practical procedure we can construct $[F_i]$ from each subproblem $P_i(y)$, simply by changing the objective coefficients in Γ_i. $[F_i]$ has either a bounded optimal solution, $x_N^* = 0$, or unbounded solutions.

Theorem 3. Assume that the degenerate linear program $[F_i]$ is solved by the perturbation method of A.Charnes. Then, if $[F_i]$ has a bounded solution, there must be at least one basic variable which has $p_j^i > 0$ in the optimal basis.

Proof: Let $x_B(\varepsilon) > 0$ be an optimal basic solution for sufficiently small $\varepsilon > 0$ and $\lim_{\varepsilon \to 0} x_B(\varepsilon) = 0$. Suppose that the objective coefficient of x_B is $p_B \leqq 0$. If $p_B \neq 0$, we have $p_B x_B(\varepsilon) < 0$. This contradicts the optimality of $x_B(\varepsilon)$, because $x = 0$ is always feasible. If $p_B = 0$, the corresponding dual solution must be identically zero, which shows that the solution can not be a feasible dual one because of some $p_j > 0$. This also contradicts the optimality.

Let $\beta^i = [\beta_N^i, E_s^i]$ be any basic matrix of $[F_i]$, where β_N^i denotes the basic columns among $\hat{A}_\Gamma^i$ and E_s^i denotes unit vectors corresponding to the slack variables in the basis. Actually, the unit vectors and β_N^i may be mixed. For the sake of simplicity, we assume they are separated as above. We shall change a present basis of P_i to a new basis which consists of the present basic variables in the rows of the basic slack variables of $[F_i]$ corresponding to E_s^i and of the new basic variables corresponding to β_N^i. In addition, we transform P_i to a canonical form for the modified new basis. We call those operations β-transformation of P_i based on β^i.

Lemma 5. An optimal basic solution to $P_i(y^*)$ and the corresponding y^* are invariant under the β-transformation.

Proof: Because every pivot row in β-transformation belongs to Γ_i, in which the basic variables attain zero.

Theorem 4. If we define a new PTM $\tilde{T}$ and the corresponding new linking parameter $\tilde{\lambda}$ as

$$\tilde{T} = T \begin{pmatrix} I_{m_o} & 0 & 0 \\ 0 & \beta^1 & 0 \\ 0 & 0 & \beta^2 \end{pmatrix}, \quad \text{and} \tag{2.8}$$

$$y = y^* + \tilde{T}\tilde{\lambda}, \tag{2.9}$$

respectively, the transformed matrix for $\tilde{\lambda}$ in the canonical form (2.3) after carrying out β-transformation based on β^i is represented as follows:

(2.10)

solution	$\tilde{\lambda}^0$	$\tilde{\lambda}^1$	$\tilde{\lambda}^2$	
$Z(y^*)$	$-\rho^0$,	$p_\beta^1 : -\rho_s^1$,	$p_\beta^2 : -\rho_s^2$,	.. criterion row
y^*	I_{m_o}	0	0	.. Γ^0
	U_0	$V_0\beta^1$	$W_0\beta^2$	
$S^1(y^*)$	U_1	$V_1\beta^1 - \alpha^1$	$W_1\beta^2$	
	0	I_{m_1}	0	.. Γ^1
	0	0	I_{m_2}	.. Γ^2
$S^2(y^*)$	U_2	$V_2\beta^1$	$W_2\beta^2 - \alpha^2$	

where p_β^i is a row vector which consists of the elements of p^i corresponding to the new entering basis β_N^i, ρ_s^i is the part of ρ^i corresponding to E_s^i, and α^i

is a matrix which consists of the remaining part of $\hat{A}^i$ for the columns corresponding to β_N^i and of zero vectors corresponding to E_s^i.

Proof: To carry out β-transformation for P_1 in (2.3) is equivalent to multiplying it by an inverse matrix H^1,

$$(2.11)\qquad H^1 = \begin{pmatrix} 1 & 0 & (\hat{c}_\beta^1, 0) \\ 0 & I & \alpha^1 \\ 0 & 0 & \beta^1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 & -(\hat{c}_\beta^1, 0)(\beta^1)^{-1} \\ 0 & I & -\alpha^1(\beta^1)^{-1} \\ 0 & 0 & (\beta^1)^{-1} \end{pmatrix},$$

where I represents an $(m-m_1)$-dimensional identity matrix. The coefficient matrix of λ is transformed to

$$(2.12)\qquad H^1 \begin{pmatrix} -\rho^0, & -\rho^1, & -\rho^2, \\ U_1 & V_1 & W_1 \\ 0 & I_{m_1} & 0 \end{pmatrix} = \begin{pmatrix} -\rho^0, & (-\hat{c}_\beta, 0)\beta^{-1} - \rho^1, & -\rho^2, \\ U_1 & V_1 - \alpha\beta^{-1} & W_1 \\ 0 & \beta^{-1} & 0 \end{pmatrix}.$$

This shows that the columns for λ^0 and λ^2 are not changed. Similarly, only the part for λ^2 is changed after β-transformation for P_2. Let $\tilde{\lambda}$ be

$$(2.13)\qquad \tilde{\lambda} = \begin{pmatrix} \tilde{\lambda}^0 \\ \tilde{\lambda}^1 \\ \tilde{\lambda}^2 \end{pmatrix} = \begin{pmatrix} I_{m_0} & 0 & 0 \\ 0 & (\beta^1)^{-1} & 0 \\ 0 & 0 & (\beta^2)^{-1} \end{pmatrix} \begin{pmatrix} \lambda^0 \\ \lambda^1 \\ \lambda^2 \end{pmatrix},$$

which is the same as in (2.9). By substituting $\tilde{\lambda}$ into the transformed matrix (2.12), we have (2.10).

$\tilde{T}$ defined by (2.8) will be denoted by $\tilde{T}_k$ at Step 4 for constructing $\tilde{D}_c(\tilde{\lambda})$, called the *second coupling problem*, in the future description of our algorithm.

Theorem 5. When $[F_i]$ has a bounded optimal solution, we have a new value of y, by solving the second coupling problem $\tilde{D}_c(\tilde{\lambda})$, which can increase the total objective value of P_λ.

Proof: $\tilde{D}_c(\tilde{\lambda})$ is represented as the dualization of (2.10) in which the signs of all elements in the objective row are reversed. It is easily seen from Theorems 3 and 4 that the present RHS of $\tilde{D}_c(\tilde{\lambda})$ has become an infeasible solution. This shows that the corresponding primal problem $\tilde{P}_c(\tilde{\lambda})$ must have an optimal solution $\tilde{\lambda}^*$ such that $\Delta Z(\tilde{\lambda}^*) > 0$ under the non-degeneracy assumption. The claim follows from Lemma 2.

Lemma 6. When $[F_i]$ is unbounded, there must be at least one slack variable in the relevant basis. Let the basis be $\beta^i = [\beta_N^i, E_s^i]$, where $E_s^i \neq \emptyset$. Let x_∞ be such a variable that the simplex criterion is negative and $\beta^{-1}\hat{a}_\infty \leqq 0$, where $\hat{a}_\infty$ denotes the coefficient vector of x_∞ in $\hat{A}_I$. Then, there must be at least one negative component of $\beta^{-1}\hat{a}_\infty$ among the rows of the basic slack variables E_s^i.

Proof. The first half of Lemma is clear from Lemma 4. We shall prove the latter half of Lemma for $[F_1]$.

$$(p_\beta,0)\beta^{-1}\hat{a}_\infty = -(\hat{c}_\beta + \rho'\beta_N,0)\beta^{-1}\hat{a}_\infty$$

$$= -(\hat{c}_\beta,0)\beta^{-1}\hat{a}_\infty - \rho'\hat{a}_\infty + (0,\rho'E_s)\beta^{-1}\hat{a}_\infty \quad . \tag{2.14}$$

The simplex criterion of x_∞ is represented as

$$0 > (p_\beta,0)\beta^{-1}\hat{a}_\infty - p_\infty = \hat{c}_\infty - (\hat{c}_\beta,0)\beta^{-1}\hat{a}_\infty + (0,\rho'E_s)\beta^{-1}\hat{a}_\infty,$$

where $p_\infty = -(\hat{c}_\infty + \rho'\hat{a}_\infty)$ denotes the coefficient of x_∞. The first two terms are non-negative because of $\hat{c}_\infty \geq 0$ and $\beta^{-1}\hat{a}_\infty \leq 0$. Consequently, the last term has to be negative. This implies that there must exist at least one negative component of $\beta^{-1}\hat{a}_\infty$ among the rows in which $\rho'E_s$ has a positive element ρ_k .

Lemma 7. If $[F_i]$ is unbounded, there exists at least one positive objective coefficient among the basic variables and x_∞.

Proof: Because $p_\beta x_B + p_\infty x_\infty$ has to attain to infinity.

Theorem 6. When $[F_i]$ is unbounded, we can always have the second coupling problem, $\tilde{D}_c(\tilde{\lambda})$, to increase exactly the objective value of p_λ .

Proof: It is obviously possible from Lemma 6 to carry out a pivot operation for entering x_∞ into the basis instead of some slack variable in the present basis. The resultant basis has at least one variable whose objective coefficient is positive from Lemma 7. For the same reason as in Theorem 5 a solution to $\tilde{D}_c(\tilde{\lambda})$ makes the objective value of P_λ increase exactly.

Corollary 1. In the unbounded case, the objective value of $P_i(y^*)$ is neither changed by β-transform nor the pivot operation adopted in Theorem 6.

Proof: Because both simplex calculations are related only to the variables in Γ_i.

The proposed algorithm is summarized as follows:

Algorithm.

Step 0. Choose $y^{(0)}$ which makes $P_i(y^{(0)})$ feasible. Put $T_0 = I$.

Step 1. Solve $P_i(y^{(k)})$ (i=1,2).

Step 2. Construct $D_c(\lambda)$ and solve it. We have λ^*, x^{i*}, $y^* = y^{(k)} + T_k\lambda^*$, and D_k . Let a new PTM be $T_k^* = T_k(D_k^{-1})'$.

Step 3. If $p^i = -(\hat{c}^i + \rho^{i\prime}\hat{A}_{\Gamma^i}) \leq 0$ for i=1,2, then the triplet (x^{1*},x^{2*},y^*) is an optimal solution to [P]. Stop.

Step 4. Solve $[F_i]$. Obtain the basis matrix β^i of $[F_i]$ in a bounded case. If it is unbounded, obtain the basis β^i after exchanging x_∞ for a basic slack variable. Transform T_k^* to $\tilde{T}_k$ by (2.8) and let a new parameter $\tilde{\lambda}$ be $y = y^* + \tilde{T}_k\tilde{\lambda}$. Then, construct $\tilde{D}_c(\tilde{\lambda})$ according to (2.10).

Step 5. Solve $\tilde{D}_c(\tilde{\lambda})$, and put $y^{(k+1)} = y^* + \tilde{T}_k\tilde{\lambda}^*$. Let a new PTM be $T_{k+1} =$

$\tilde{T}_k(\tilde{D}_k^{-1})'$, where $\tilde{D}_k$ denotes the optimal basis matrix of $\tilde{D}_c(\tilde{\lambda})$. Go to Step 1, regarding $\tilde{\lambda}$ as λ in P_λ.

Notes:

1) We may choose another basis in $[F_i]$ that has as many positive p_j^i as possible. If it might be possible, the algorithm would be more efficient by Theorem 4.

2) We can return to Step 3 at Step 5, by changing the basis of P_i to the new basis obtained by $[F_i]$ in Step 4. It has been seen by our computational experience that the amount of computation for this procedure with some *additional routines* is much less than that for the algorithm above, although the procedure becomes more complex.

3) A multi-stage case can be dealt with likewise, if the total number of linking variables is within the limitation on the number of rows of the LP subroutine built in the system for solving the coupling problems.

Theorem 7. *The proposed algorithm terminates in a finite number of steps* under the non-degeneracy assumption.

Proof: It has been proved in Theorems 5 and 6 that the objective value of P_λ is always increased by the change of $y^{(k)}$. Moreover, the new basis matrix of P_λ for $y^{(k+1)}$ which is represented as a pair of the optimal basis matirces for $P_1(y^{(k+1)})$ and $P_2(y^{(k+1)})$ must be different from the basis matrix chosen for $y^{(k)}$ in the previous cycle because of Lemma 3. This implies a finite termination of the algorithm, because there are only a finite number of pairs of possible basis matrix for P_λ in a course of selecting $y^{(k)}$ in the algorithm.

3. Computational Experience.[5)]

Four types of models were used for our test problems, in which Models I and II were for simple refinery production planning and Models III and IV were cutting stock problems of Eiseman's type. The total number of cases solved is 32, in which the types of some constraints, and their coefficients and RHS's are somewhat changed according to the assumed situations. Models I and II were adopted as typical examples of problems in a great deal of real use and it is relatively easy to estimate the tendency of the values of linking variables in these problems. The cutting problems(III & IV) were adopted as an example such that the optimal basis and the values of linking variables are very much sensitive

5) The author is much indebted to Mr. M. Morita, his student at Kobe University of Commerce, for helping to generate the test problems and also for computer programming, and also to Professors L.S. Lasdon, Case Western Reserve University and R.E. Marsten, Massachusetts Institute of Technology, for releasing SEXOP program for us.

to the pattern of demand, i.e., the figures in RHS.

A test version of the algorithm, named PAIROP, was written in FORTRAN using SEXOP (Subroutines for Experiemntal Optimization [12]) for HITAC 8250 Computer with 64K core storage and slow-speed auxiliary storage. Since the original SEXOP, which was released by Professor R.E. Marsten for us, was too large for our computer above and could not be maintained in core, it was operated by overlay between core storage and auxiliary memory, and the dimensions for the working data area were considerably reduced.

Table 1

Structure of the Test Problems*

Model	No. of Cases	m	n	Density
I	3	9	16	21.5 %
II	4	10	20	19.5
III	20	8	20	33.2 (1st period)
			14	24.1 (2nd period)
IV	5	8	20	33.2

* All problems are of 2-period and the submodel in each period has the same size except Model III.

** m = the number of rows in a period; n = the number of columns (including slacks); the density is for each submodel (excluding linking variables).

The rate of convergence can be investigated by referring to the number of times of solving the coupling problems, $D_c(\lambda)$ and $\tilde{D}_c(\tilde{\lambda})$, which is also called *the number of cycles* required for optimality by the algorithm. The number of cycles is shown in Table 2. For the sake of simplicity, in all cases for Models III and IV the initial values of linking variables were given as zero. The optimal solution and the corresponding basis varied much in every case. However, it should be noticed in Table 2 that the number of cycles is mostly one or two, which is unexpectedly small because, roughly speaking, it may be considered that solving the coupling problem once corresponds to the generation of each column for the subproblems in a column-generation method. This may show that the algorithm would work very efficiently even for larger real problems if a good initial value for the linking variable could be chosen. Though it was supposed that solving the second coupling problem would require more time because of the extra operations concerning β-transformation, the results in Table 2 show that the computing time is almost proportional to the sum of both numbers of times of solving $D_c(\lambda)$ and $\tilde{D}_c(\tilde{\lambda})$ respectively. If let the number of cycles required for optimality be k and the computing time be t, then we have $t = 15.0 + 7.9k$ ($r^2 = 0.95$) for Model I & II and $t = 12.2 + 7.2k$ ($r^2 = 0.96$) for Model III & IV.

In order to compare our algorithm with Beale's method [4] in computational efficiency, we solved all 25 cases for Model III & IV again by restricting the entering basis in the direction-finding problem to only one variable[6], which we call Beale-like method hereafter. Among them, four cases were different in both the computing time and the number of cycles required for optimality, and the others were almost the same in computing time and completely the same in the number of cycles. The results for four cases above appear in Table 3, in which our method is found to be 19% to 28% faster than the Beale-like method. It was observed that in those cases the plural number of variables with a positive p_j entered into the basis in the direction-finding problem, and, in other cases, only one variable with a positive p_j entered into the basis even though several variables were exchanged in the direction-finding problem. This result seems to endorse the note (1) mentioned below the description of our algorithm in Section 2.

Table 2

Number of Cycles required for Optimality

Number of Cycles	1	2	3	4	Average
1) Model I & II					
Number of Cases	4	2	0	1	2.4
Average CPU Computing Time* (sec.)	23.75	36.00	-	72.00	34.1
2) Model III & IV					
Number of Cases	7	13	4	1	2.9
Average CPU Computing Time* (sec.)	19.14	34.23	47.25	64.00	33.3

* As an example of comparison with a direct simplex method, LPS/NDOS for HITAC 8250 (Linear Programming System under New Disc Operating System in the Library) can be referred to. According to the Manual (Hitachi Co.,August 1973), the computing time for a nutrition problem of 13 x 21 size with a matrix of 26% density is reported to be 41 seconds for 18 iterations.

Tables 4 and 5 show the estimated core storage requirements for the conventional simplex method and our algorithm respectively. It can be seen that our algorithm requires considerably less core storage than the conventional simplex method.

6) See the description concerning the Beale-like method mentioned in Section 1.

Table 3

Comparison of the Algorithm with Beale-like Method

	Our Algorithm		Beale-like Method	
Cases	Number of Cycles	Computing Time	Number of Cycles	Computing Time
No.1	2 times	38 sec.	3 times	53 sec.
No.2	2	36	3	50
No.3	3	51	4	63
No.4	2	31	3	38

Table 4

Core Storage Requirements for Conventional Simplex(words)*

	Dimensions	Model IV (m=10,n=20)
Matrix	$2m(m+2n)$	1000
RHS	$2m$	20
Dual Variables	$2m$	20
Cost	$m+2n$	50
Primal Variables	$m+2n$	50
Total	$2m^2+4mn+6m+4n$	1140

Table 5

Core Storage Requirements for Our Algorithm(words)*

	Dimensions	Model IV (m=10,n=20)
Matrix	$3m^2$	300
RHS	m	10
Cost	$3m$	30
Inverse	m^2	100
Dual Variables	m	10
PTM	m^2	100
Linking Variables	m	10
Primal Variables	$3m$	30
Total	$5m^2+9m$	590

* Neglecting the working storages and minor items.

4. An Interactive Approach to a Preference Optimization.

We shall consider a case in which a decision-maker wishes to optimize his own implicit preference funciton $U(z^1,z^2)$, structuring his preference attitude for each pair (z^1,z^2). It is supposed that the preference function depends mainly on the decision-maker's feeling for uncertainty in the future. It is

reasonably assumed that $U(z^1,z^2)$ is increasing in each z^i.

For example, let us consider a case such that there are two programs: A with the pair (z_a^1,z_a^2) and B with the pair (z_b^1,z_b^2) on a given line $z^1 + z^2 = z$-goal, where we assume $z_a^1 << z_b^1$ and $z_a^2 >> z_b^2$. The decision-maker may wish to increase z^1 much more at A than the unreliable z^2. On the other hand, he may wish to increase z^2 much more at B.

Geoffrion, Dyer and Feinberg [7] presented an interactive method to find a decision-maker's implicit utility function defined on multi-criteria. The method is equivalent to determining a weight α at each observed point (z^1,z^2) such that $z = z^1 + \alpha z^2$, where α may be the ideal marginal proportion of change for two subobjectives in this case. Our proposed algorithm has such a feature that we can adopt the interactive procedure easily.

We can separate the total objective Z into two subobjectives z^1 and z^2 in P_λ as follows:

$$z^1(x^1:\lambda) = -\hat{c}^1x_N^1 - \Delta_1'\lambda \ , \text{ and } \quad z^2(x^2:\lambda) = -\hat{c}^2x_N^2 + \Delta_2'\lambda \ ,$$

where $\Delta_i = \pi^{i\prime}T$ and $\rho = \Delta_2 - \Delta_1$. The coupling problem in this case is defined on the basis of $\rho + (\alpha-1)\Delta_2$ instead of ρ. The objective functions of the direction-finding problems $[F_i]$ $(i=1,2)$ can be represented as $\max\ [p^1 + (1-\alpha)\Delta_2^1A_\Gamma^1]x_N^1$ for $[F_1]$, and $\max\ [p^2 + (1/\alpha-1)\Delta_1^2A_\Gamma^2]x_N^2$ for $[F_2]$, respectively, where Δ_i^k denotes the component of Δ_i corresponding to Γ_k. Examining the pair of the subobjective values $(z^1(y),z^2(y))$, the new value of y is obtained by the ordinary procedure in the interactive method, and the process is repeated. It is known that the process converges to the optimum under some condition [7], but we think that this approach is more siginificant in a case such that the optimization process must be stopped without attaining to an optimum. Especially, such cases may commonly arise in the actual planning in a large system. Unless such a case, it will be more practical to solve firstly an optimum for a predetermined value of α and then carry out a parametric analysis on α for the optimal solution. This solution-method corresponds to an extension of our algorithm to the case of the non-linear concave objective function $U(z^1,z^2)$ which is increasing in each z^i.

Acknowledgement. The author is grateful to the referees and Professor Hiroshi Watanabe, the University of Tsukuba, for comments and suggestions which have results in a much improved version of the paper.

References

[1] Aonuma,T.: Two-level Approach to Dynamically Decomposing a Linear Planning Model. Working Paper No.26(September 1975),The Institute of Economic Research, Kobe University of Commerce.

[2] Aonuma,T.: A Two-level Algorithm and Hierarchically Decomposed Submodels for Multi-period Planning — Numerical Examples — (in Japanese). *Journal of Kobe University of Commerce*, Vol.28, No.1(1976), 7-27.

[3] ___"___ : A Nested Multi-level Planning Approach to a Multi-period Linear Planning Model. Working Paper No.35(January 1977), The Institute of Economic Research, Kobe University of Commerce.

[4] Beale,E.M.L.: The Simplex Method Using Pseudo-basic Variables for Structured Linear Programming Problems. *Recent Advances in Mathematical Programming (ed. R.Graves and P.Wolfe)*. McGraw-Hill, New York, 1963, 133-148.

[5] Dantzig,G.B.,and Wolfe,P.: Decomposition Principle for Linear Programs. *Operations Research*, Vol.8(1960), 101-111.

[6] Geoffrion,A.M.: Primal Resource-directive Approaches for Optimizing Nonlinear Decomposable Systems. *Operations Research*, Vol.18(1970), 375-403.

[7] Geoffrion,A.M., Dyer,J.S., and Feinberg,A.: An Interactive Approach for Multi-criterion Optimization, with an Application to the Operation of an Academic Department. *Management Science*, Vol.19(1972), 357-368.

[8] Glassey,C.R.: Nested Decomposition and Multi-stage Linear Programs. *Management Science*, Vol.20(1973), 282-292.

[9] Grinold,R.C.: Steepest Ascent for Large Scale Linear Programs. *SIAM Review*, Vol.14(1972), 447-464.

[10] Heal,G.M.: *The Theory of Economic Planning*. North-Holland, Amsterdam,1973.

[11] Ho,J.H., and Manne,A.S.: Nested Decomposition for Dynamic Models. *Mathematical Programming*, Vol.6(1974), 121-140.

[12] Marsten,R.E.: Users Manual for SEXOP. Sloan School of Management, Massachusetts Institute of Technology, February 1974.

[13] Mesarovic,M.D., Macko,D., and Takahara,Y.: *Theory of Hierarchical, Multilevel, Systems*. Academic Press, New York, 1970.

[14] Simonnard,M.: *Linear Programming*. Prentice-Hall, New York, 1966.

[15] Wismer,D.A. (ed.): *Optimization Methods for Large-scale Systems*. McGraw-Hill, New York, 1971.

Tatsuo AONUMA: Department of Management Sciences, Kobe University of Commerce, Tarumi, Kobe, 655, Japan.