International Institute for
Applied Systems Analysis
IIASA www.iiasa.ac.at

Decomposition Algorithm Based on the Primal-Dual Approximation

Nurminski, E.A.

IIASA Working Paper

May 1982

Working Paper

DECOMPOSITION ALGORITHM BASED ON
THE PRIMAL-DUAL APPROXIMATION

E.Nurminski

May 1982
WP-82-46

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

DECOMPOSITION ALGORITHM BASED ON THE PRIMAL-DUAL APPROXIMATION

E.Nurminski

May 1982
WP-82-46

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

Decomposition algorithm based on the primal-dual approximation

E. Nurminski

International Institute for Applied Systems Analysis,

Laxenburg, Austria

ABSTRACT

A decomposition algorithm based on the simultaneous approximation of the primal and dual forms of an optimization problem is proposed. This approach makes maximum use of the primal-dual information available during solution of the decomposed problem, speeds up the convergence, and provides upper and lower bounds for the optimum.

1. Introduction

This paper presents an extension of the approach considered in [1]. The new proposed decomposition algorithm is based on the simultaneous approximation of the primal and dual forms of structured optimization problems.

Consider the following two-block mathematical programming problem with linking variables:

$$\begin{aligned} &\min \ \{ \ c_A(z_A) + c_B(z_B) \ \} \\ &g_A(z_A, x) \le 0 \\ &g_B(z_B, x) \le 0 \end{aligned} \tag{1}$$

where z_A and z_B can be viewed as internal variables of subproblems

$$f_A(x) = \min_{g_A(z_A,x) \le 0} c_A(z_A) \quad , \quad f_B(x) = \min_{g_B(z_B,x) \le 0} c_B(z_B)$$

and the corresponding optimal values $f_A(x), f_B(x)$ are functions of linking variable x. We denote the euclidian space of linking variables by E and the space of real numbers by R.

Problem (1) can then be redefined as the problem of finding

$$v^* = \min_x \{ f_A(x) + f_B(x) \} \tag{2}$$

We shall refer to this as the primal form of problem (1).

If c_A, c_B are convex with respect to z_A, z_B and g_A, g_B are jointly convex with respect to the pairs (z_A, x) and (z_B, x), then $f_A(x)$, $f_B(x)$ are convex functions. This allows the use of convex analysis in studying the convergence properties of the proposed algorithms; some particular notions used are summarized below.

We denote the subgradient set of convex function $f(x)$ at x by $\partial f(x)$:

$$\partial f(x) = \{ g : f(y) \ge f(x) + g(y - x) \}$$

where $g(y - x)$ represents the inner product of vectors g and $y - x$. Generally we will denote the inner product of two vectors x and y by $x\,y$.

The conjugate of a convex function $f(x)$ is denoted by $f^*(p)$:

$$f^*(p) = \sup_x \{ px - f(x) \} = x_p p - f(x_p)$$

The application of standard convex duality theory to problem (2) leads to the following equality:

$$\min_x \{ f_A(x) + f_B(x) \} = -\min_p \{ f_A^*(-p) + f_B^*(p) \} \tag{3}$$

and we shall refer to the right-hand side of this equality as the dual form of problem (1).

The algorithm proposed in [1] for solving problem (2) is based on the idea of replacing (2) by the sequence of problems

$$\min_{x} \{ f_A(x) + f_B(x)^k \} \tag{4}$$

where $f_B(x)^k$ is the approximation of the function $f_B(x)$ obtained on the k-th iteration.

It was suggested that this approximation should be derived by constructing a piece-wise linear support function for $f_B(x)$ based on the values of this function and its subgradient ∂f_B already computed. This approximation is gradually refined, directing the sequence of solutions of the auxiliary problems (4) toward the solution of the problem (2).

The resulting algorithm performs quite satisfactorily but does not make full use of the information available during the optimization process. Another drawback is that it does not produce both upper and lower estimates of the optimum, which makes it difficult to determine the rate of convergence.

Also, in many practical cases, information on the solution of the dual form of problem (2), which can be interpreted as a set of shadow prices for linking variables, may provide additional insight into the qualitative properties of problem (1). This information is not readily available even if the solution of the primal form is known; substantial further analysis of the problem is required to reveal it. Similarly, if the algorithm described in [1] is applied to the dual form of problem (1) then the primal solution cannot be found immediately.

The proposed extension is based on the simultaneous use of approximation in the primal and dual formulations of problem (2) and leads to the algorithm discussed in Section 2. This algorithm provides both primal and dual solutions of problem (3), supplies upper and lower estimates of the optimum during solution, and, as the numerical experiments will show, converges sufficiently rapidly.

To investigate the convergence of this algorithm, we need to examine the interrelations between the convergence of sequences of convex functions and the convergence of their conjugates. From the point of view of this study, the most interesting and useful case is when the convergence of one of these sequences implies the convergence of the other. This type of convergence is defined in terms of the convergence of epigrafs of convex functions.

Definition. The epigraf $epi(f)$ of a convex function $f(x)$ is a subset of an extended space $R \times E$ such that the pair (μ, x) belongs to $epi(f)$ if and only if $\mu \geq f(x)$.

Definition. A sequence $\{ f_n(x) \}$ of convex functions is called e-convergent toward $f(x)$ if

$$\lim_{n \to \infty} epi(f_n) = epi(f)$$

Convergence of sets is defined as follows:

$$\bigcap_n \bigcup_{m \geq n} epi(f_m) \equiv \overline{\lim_{n \to \infty}}\, epi(f_n) = \underline{\lim_{n \to \infty}}\, epi(f_n) \equiv \bigcup_n \bigcap_{m \geq n} epi(f_m)$$

We will denote e-convergence by

$$e\text{-}\lim_{n \to \infty} f_n(x) = f(x)$$

Note that e-convergence implies point-wise convergence, but the converse is not true.

The importance of e-convergence in this study is based on the fact that if

$$e\text{-}\lim_{n \to \infty} f_n(x) = f(x)$$

then

$$e\text{-}\lim_{n \to \infty} f_n^*(x) = f^*(x)$$

so that the conjugate functions are also point-wise convergent. The historical background and a general statement of this result can be found, for instance, in [2].

We are interested in a special case of this result for monotone sequences of functions $\{ f_n(x) \}$ for which the theory is simpler than in the general case. Under conditions of monotonicity there is generally no difference between point-wise and e-convergence, and so we can now use theory relating to e-convergence in the following sections without further comment.

2. Theory

Consider an algorithm with the following structural form:

BEGIN *PRIMAL-DUAL DECOMPOSITION ALGORITHM*

Let $k = 0$. At the initial point p^0 the value of the function $f_B^*(p)$ is $f_B^*(p^0)$. Define the initial approximations $f_B(x)^{-1}$ and $f_A^*(p)^{-1}$ as

$$f_B(x)^{-1} \equiv \infty; \; f_A^*(p)^{-1} \equiv \infty$$

While *(NOT SOLUTION)*

BEGIN *INNER LOOP*

Using $f_B^*(p^k), p^k$ update the approximation of the function $f_B(x)$:

$$f_B(x)^k = \max \{ f_B(x)^{k-1} , xp^k - f_B^*(p^k) \} \tag{5}$$

Solve the auxiliary problem

$$\min_x \{ f_A(x) + f_B(x)^k \} = f_A(x^k) + f_B(x^k)^k = v_k \tag{P}$$

Update the approximation of the function $f_A^*(-p)$:

$$f_A^*(-p)^k = \max \{ f_A^*(-p)^{k-1} , -x^k p - f_A(x^k) \} \tag{6}$$

Solve the auxiliary problem

$$\min_{p} \{ f_A^*(-p)^k + f_B^*(p) \} = f_A^*(-p^{k+1})^k + f_B^*(p^{k+1}) = -w_k \tag{D}$$

Set $k = k+1$

END *INNER LOOP*

END *PRIMAL-DUAL DECOMPOSITION ALGORITHM*

For the algorithm to be well-defined it is necessary that auxiliary optimization problems (P) and (D) have finite solutions. We can guarantee this by requiring that the functions $f_A(x)$, $f_B(x)$ and their conjugates $f_A^*(p)$, $f_B^*(p)$ are finite. This is rather a strong assumption, but we shall adopt it for the time being to simplify the theoretical considerations.

It is clear from the way we have constructed the functions $f_B(x)^k$, $f_A^*(-p)^k$ that they have the following properties:

$$f_B(x)^k \le f_B(x)$$

$$f_A^*(-p)^k \le f_A^*(-p)$$

and are therefore finite convex functions if the right-hand sides of the above inequalities are finite. We therefore have the finite point-wise limit functions:

$$\lim_{k \to \infty} f_B(x)^k = f_B(x)^\infty$$

$$\lim_{k \to \infty} f_A^*(-p)^k = f_A^*(-p)^\infty$$

and the monotone character of these sequences means that they are also e-convergent.

The monotonicity of the sequences also means that the following limits exist:

$$\lim_{k \to \infty} v_k = v, \quad \lim_{k \to \infty} w_k = w$$

The theoretical validity of the algorithm is based on the following theorem.

Theorem. If sequences $\{x^k\}$, $\{p^k\}$ are bounded then any limit points x', p' of these sequences are solutions of the primal and dual forms, respectively, of problem (2).

Proof. Passing to the limit in (6) leads to

$$f_A^*(-p)^\infty \geq -x'p - f_A(x')$$

for any p, or

$$f_A(x') \geq \{ f_A^*(-p)^\infty \}^*(x')$$

However, in general

$$f_A(x) \leq \{ f_A^*(-p)^\infty \}^*(x)$$

so for $x = x'$, the equality

$$f_A(x') = \{ f_A^*(-p)^\infty \}^*(x') \tag{7}$$

holds.

By similar arguments, passing to the limit in (5) leads to

$$f_B^*(p') = \{ f_B(x)^\infty \}^*(p') \tag{8}$$

The e-convergence of the sequence $\{ f_B(x)^k \}$ implies that

$$f_B^*(p') = f_B^*(p')^\infty$$

and, similarly, from the e-convergence of the sequence $\{ f_A^*(p)^k \}$

$$f_A(x') = f_A(x')^\infty$$

The rest is easy. Let

$$v = \lim_{k \to \infty} \min_x \{ f_A(x) + f_B(x)^k \} = \min_x \{ f_A(x) + f_B(x)^\infty \} =$$

$$f_A(x') + f_B(x')^\infty = f_A(x') + f_B(x')$$

Also

$$-w = \lim_{k \to \infty} \min_{p} \{ f_A^*(-p)^k + f_B^*(p) \} = \min_{p} \{ f_A^*(-p)^\infty + f_B^*(p) \} =$$

$$f_A^*(-p')^\infty + f_B^*(p') = f_A^*(-p') + f_B^*(p')$$

By construction

$$v \leq w$$

However, for any p

$$v = \min_{x} \{ f_A(x) + f_B(x)^\infty + px - px \} \geq$$

$$\min_{x} \{ f_A(x) + px \} + \min_{x} \{ f_B(x)^\infty - px \} =$$

$$-f_A^*(-p) - \{ f_B(x)^\infty \}^*(p)$$

Hence for $p = p'$

$$v \geq -f_A^*(-p') - \{ f_B(x)^\infty \}^*(p') = -f_A^*(-p') - f_B^*(p') =$$

$$-f_A^*(-p')^\infty - f_B^*(p') = -\min_{p} \{ f_A^*(-p)^\infty + f_B^*(p) \} = w$$

This demonstrates that

$$v = w = v^*$$

and the theorem is proved.

To complete the theoretical discussion of the algorithm we should mention the related work of K. Aneros. The algorithm proposed in [3] combines primal and dual approximations and, rather than solving them separately as problems (P), (D), incorporates them into one optimization problem. This differs from the algorithm considered in this paper but the general idea of using a primal-dual approximation is somewhat similar.

3. Examples

The algorithm was implemented using the code MINOS [4] to solve the auxiliary linear problems. Unfortunately, MINOS does not have special subroutines capable of modifying the internal representation of the data when the parameters of the problem are changed or when additional rows/columns are added, and for this reason the auxiliary subproblems must be formulated and updated through modification of the input files.

This is clearly not the most efficient way to implement the algorithm, but at this stage we are more concerned with the number of major iterations required than with computational efficiency as a whole.

One advantage of this approach was the small amount of additional programming needed to supply codes for generating updated input files. Some of the UNIX functions [5] proved very useful in this respect.

The chosen mode of implementation also resulted in some loss of accuracy, as will be seen later.

Our first example concerns the Polish agricultural model developed for the Food and Agriculture program at IIASA [6]. The detailed structure of the model is described elsewhere; here we consider this model only as a subject on which to test the proposed decomposition algorithm.

For test purposes this model was decomposed into two submodels (MIT and MID), some characteristics of which are given below.

MID:

	total	normal	free	fixed	bounded
rows	174	46	9	115	4
columns	249	231	0	1	17

no. of matrix elements 2331 density 5.380

MIT:

	total	normal	free	fixed	bounded
rows	171	57	7	105	2
columns	240	219	0	1	20

no. of matrix elements 2096 density 5.107

These two subproblems are linked by a group of 11 constraints which represent either the distribution of common resources between submodels or the balancing of certain flows between submodels. The linking constraints were transformed into linking variables by introducing an additional set of linking variables, each one corresponding to the value of a linking row.

This problem was solved using the algorithm described above and the results were compared with those obtained using the earlier algorithm [1], which is based on approximation only in the primal form of the problem. In this experiment subproblem MID was used in its primal form (as $f_A(x)$) and MIT in its dual form (as $f_B^*(p)$). The results are given in Tables 1 and 2.

Table 1. Convergence of the primal algorithm.

cycle	iter	upper est	iter	MID optimum
1	277	0.137265d+09	437	-0.326988d+05
2	77	-0.158735d+06	60	0.238563d+08
3	3	-0.158735d+06	37	-0.310690d+05
4	3	-0.158735d+06	42	-0.260705d+05
5	4	-0.158740d+06	10	-0.249800d+05
6	3	-0.158823d+06	1	-0.248799d+05

Table 2. Convergence of the primal-dual algorithm.

cycle	iter	upper est	iter	lower est
1	277	0.137265d+09	444	-0.326988d+05
2	77	-0.158735d+06	89	-0.513099d+07
3	3	-0.158735d+06	49	-0.164955d+06
4	3	-0.158735d+06	32	-0.159957d+06
5	4	-0.158740d+06	76	-0.158866d+06
6	3	-0.158823d+06	26	-0.158826d+06

The primal algorithm was applied to the dual of (1) and consequently produced only an upper estimate of the optimal value, which is shown in Table 1 together with the number of simplex iterations required to solve the auxiliary optimization problem. The optimal value obtained in subproblem MID is also shown in Table 1, together with the number of simplex iterations required to obtain this value. It was possible to use the optimal solution obtained on a previous major iteration as a starting basis for the next cycle. As a result, the number of auxiliary simplex iterations decreases rapidly as the algorithm progresses. Table 1 also illustrates the nonmonotone behavior of the optimal value obtained for the MID subproblem.

Table 2 shows the results obtained with the primal-dual algorithm, which provides both upper and lower estimates of the optimum; again, the numbers of simplex iterations required to solve the auxiliary optimization problems are also given.

When solving the auxiliary optimization problem (D) it was again possible to use the preceding optimal solution as a starting point for each new cycle; Table 2 shows the rapid decrease in the number of additional simplex iterations required to reach the optimum.

For the problem (P), however, the previous optimal solution is not feasible so it was just used as an advanced starting basis for the next iteration . Table 2 shows that this also leads to a substantial decrease in the numbers of additional simplex iterations.

It is also worth noting that the Dantzig-Wolfe algorithm took 49 iterations to achieve 3-digit accuracy when applied to this problem.

The following table demonstrates the actual solution of this problem.

Table 3. Solution of the agricultural model.

var	primal-dual algorithm activity	primal-dual algorithm shadow price	primal algorithm activity	primal algorithm shadow price
1	0.238889d+04	0. d+00	0.238889d+04	0. d+00
2	0.238889d+04	0.324081d+00	0.238889d+04	0.324081d+00
3	0.574402d+04	0.132165d-01	0.574402d+04	0.132165d-01
4	0.193036d+03	0. d+00	0.193036d+03	0. d+00
5	0.142345d+04	0. d+00	0.142345d+04	0. d+00
6	0.335377d+03	0. d+00	0.335377d+03	0. d+00
7	0.145165d+03	0. d+00	0.145165d+03	0. d+00
8	0.463994d+03	0. d+00	0.463994d+03	0. d+00
9	0.295227d+05	0.146246d+00	0.295227d+05	0.146246d+00
10	0. d+00	0. d+00	0. d+00	0. d+00
11	0.989897d+01	-0.666134d-15	0. d+00	-0.222045d-15

In this case the primal algorithm was applied to the dual of the original problem and consequently produced a dual solution which agrees well with the solution provided by the primal-dual algorithm.

The column headed "activity" for the primal algorithm gives the values of linking variables generated by the subproblem MID in response to the optimal prices. It is interesting to note that the value calculated for primal variable number 11 differs from the optimal value obtained by the primal-dual algorithm. This demonstrates that price information alone is insufficient to calculate the overall optimum. However, this is the only variable which should be controlled directly; the optimal values for the rest of the variables may be obtained using the optimal price information.

One of the shortcomings of the primal algorithm is that its performance depends strongly on which one of the subproblems is approximated (as $f_B(x)$), and which is taken in full (as $f_A(x)$). In the worst case this can cause a marked deterioration in the performance of the algorithm. The primal-dual algorithm allows both subproblems to be considered in their unapproximated form (either primal or dual) and so it may be less sensitive to the roles assigned to the subproblems. To investigate this hypothesis the experiment described above was repeated using the primal form of subproblem MIT (as $f_A(x)$) and the dual form of subproblem MID (as $f_B^*(p)$). The primal algorithm was stopped after executing 37 major iterations and the primal-dual algorithm after performing 20 major iterations. The results are given in Table 4.

Table 4. Convergence of primal-dual and primal algorithms.

cycle	primal-dual algorithm				primal algorithm	
	iter	lower est	iter	upper est	iter	upper est
1	420	-0.13093d+06	386	0.27334d+09	386	0.27334d+09
5	422	-0.10270d+08	47	0.34157d+07	32	-0.11457d+06
10	9	-0.16912d+06	86	-0.15348d+06	76	-0.15290d+06
15	20	-0.15987d+06	17	-0.15816d+06	16	-0.15691d+06
20	5	-0.15885d+06	66	-0.15880d+06	32	-0.15854d+06
25		-		-	23	-0.15880d+06
30		-		-	17	-0.15881d+06
35		-		-	19	-0.15882d+06
36		-		-	27	-0.15882d+06
37		-		-	18	-0.15882d+06

The computational process is illustrated in Figure 1. This graph shows the convergence of the upper and lower bounds for the primal-dual algorithm (curves 0 and 1), and of the upper bound for the primal algorithm (curve 2). The figure illustrates the relative accuracy (on a logarithmic scale) of each bound, which is calculated as

$$y_i = \frac{|f_i - f^*|}{f^*}$$

where f_i denotes the value obtained for the objective function on the i-th cycle and f^* denotes the optimal value (here taken to be $-0.158824d+06$). It is clear that the primal-dual algorithm converges slightly faster in the range of relative accuracy from 10^{-2} to 10^{-4}.

The second example concerns a simplified version of the revised energy model MESSAGE (Model for Energy Supply Systems Alternatives and their

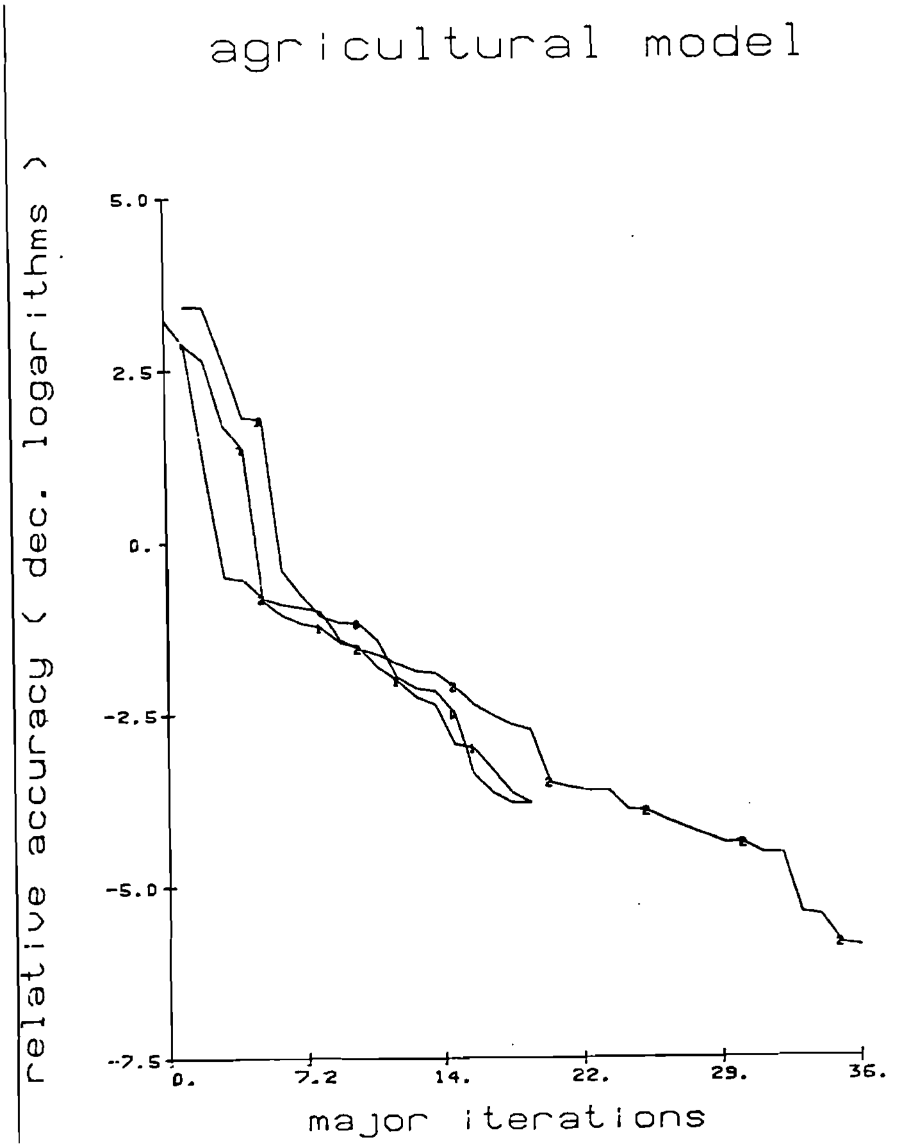

Figure 1. Convergence of the primal-dual and the primal algorithms.

General Environmental impact) [7] which is described in more detail in [8]. This is a dynamic linear programming model intended to describe the dependence of the transition from one pattern of energy production to another on the availability of certain resources and on environmental effects.

The simplified version describes the production of energy from various raw materials and its transportation, distribution and conversion to meet a final demand specified outside the model.

For test purposes this model was decomposed into 2 submodels and solved using the algorithm described above.

The first submodel (CENTR) describes the production of different kinds of final energy from fossil and nuclear fuels, hydropower plants, solar installations, geothermal plants and other sources. The final energy is produced in the form of electricity, district heat, hydrogen, coal, and liquid and gaseous fuels.

The second submodel (END) deals with the transformation of final energy into useful energy, and describes the flow of final energy through the different stages of transportation, distribution, and on-site conversion to meet the demand of end-users.

The characteristics of these submodels are as follows:

CENTR:

	total	normal	free	fixed	bounded
rows	246	182	22	42	0
columns	202	192	0	4	6

no. of matrix elements 963 density 1.938

END:

	total	normal	free	fixed	bounded
rows	157	102	13	42	0
columns	139	126	0	3	10

no. of matrix elements 520 density 2.383

The linking variables in this model are the flows of final energy between subproblems; different variants of the model can be specified which differ in the number of time periods considered, number of technologies represented, and so on. For this test, the number of links between subsystems was chosen to be 42, which corresponds to 7 time periods. A number of simplifications have also been made in the structure of the subproblems to cut down the size of the blocks.

In this experiment subproblem END was used in its primal form (as $f_A(x)$) and CENTR in its dual form (as $f_B^*(p)$).

The results obtained with the proposed algorithm are shown in Table 5.

Table 5. Convergence of the primal-dual algorithm.

cycle	iter	upper est	iter	lower est
1	188	0.110820d+05	85	0.168162d+04
2	19	0.943634d+04	17	0.822472d+04
3	4	0.934955d+04	10	0.910782d+04
4	1	0.934441d+04	10	0.914509d+04
5	5	0.928744d+04	7	0.923855d+04
6	5	0.926119d+04	6	0.925425d+04
7	5	0.925607d+04	2	0.925605d+04

The results given in Table 5 correspond to those listed in Table 2, but in this case the algorithm stopped at the 7-th major iteration (cycle) due to rounding errors. It is clear that to obtain more precise results it would be necessary to improve the accuracy of the information passed between subproblems.

Table 6 gives the results obtained for the same problem with the primal decomposition algorithm.

Table 6. Convergence of the primal algorithm.

cycle	iter	upper est	iter	END optimum
1	188	0.110820d+05	85	0.168162d+04
2	19	0.943634d+04	17	0.723144d+04
3	1	0.934919d+04	7	0.821608d+04
4	1	0.934909d+04	3	0.831283d+04
5	4	0.930227d+04	5	0.822231d+04
6	1	0.929622d+04	3	0.836067d+04
7	3	0.925830d+04	1	0.833766d+04
8	1	0.925781d+04	3	0.837510d+04
9	1	0.925604d+04	1	0.837609d+04

To test the robustness of the algorithm the experiment was again repeated, this time using the primal form of subproblem CENTR (as $f_A(x)$) and the dual form of subproblem END (as $f_B^*(p)$). Both algorithms were stopped after executing 55 major iterations.

The computational process is illustrated in Figure 2. This graph shows

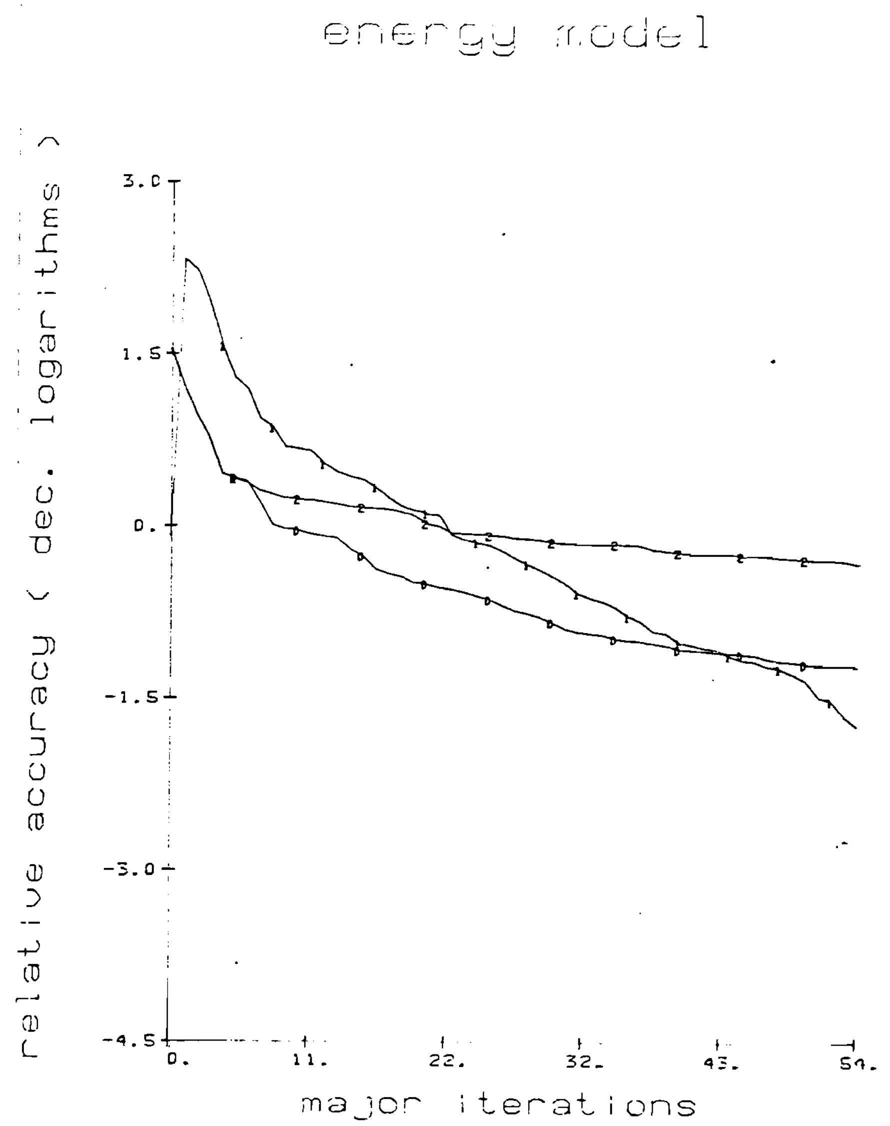

Figure 2. Convergence of the primal-dual and the primal algorithms.

the convergence of the upper and lower bounds for the primal-dual algorithm (curves 0 and 1), and of the upper bound for the primal algorithm (curve 2). The figure illustrates the relative accuracy (on a logarithmic scale) of each bound, which is calculated as before as

$$y_i = \frac{|f_i - f^*|}{f^*}$$

where f_i denotes the value obtained for the objective function on the i-th cycle and f^* denotes the optimal value (here taken to be 0.925606d+04). It is clear that the primal-dual algorithm converges slightly faster then the primal algorithm although it was unable to attain a relative accuracy of more then 10^{-2} in the given number of major iterations.

4. Concluding remarks

This limited trial of the primal-dual algorithm shows that it has some advantages over the earlier decomposition algorithm in dealing with structured problems. Nevertheless, it is clear that further work is required on both theoretical and practical aspects of this approach.

One interesting theoretical development would be to investigate the use of this algorithm for the piece-wise linear case. Another important development would be the extension of the algorithm to pathological (unbounded or infeasible) problems.

From the practical point of view, the efficiency of the implementation could be improved, firstly by ensuring that more accurate information passes between subproblems and, secondly, by providing more advanced means for modifying the subproblems.

Acknowledgments

The author wishes to thank P.O. Lindberg for useful discussions and for drawing his attention to the work of K. Aneros.

References

1. E. Nurminski, "Convergence and Numerical Experiments with a Decomposition Algorithm," RR-81-31, International Institute for Applied Systems Analysis, Laxenburg, Austria (1982).

2. R.J.-B. Wets, "Convergence of Convex Functions, Variational Inequalities and Convex Optimization Problems," pp. 376-403 in *Variational Inequalities and Complementarity Problems*, ed. R.W. Cottle, F. Giannessi, and J-L. Lions,Wiley, Chichester (1980).

3. K. Aneros, "A General Framework for Solving Structured Linear Programs by Column and Row Generation Techniques," TRITA-MAT-1978-9, Department of Mathematics, Royal Institute of Technology, Stockholm (1978).

4. B.A. Murtagh and M.A. Saunders, "MINOS. A Large-Scale Nonlinear Programming System (for Problems with Linear Constraints)," Technical report SOL 77-9, Stanford University (1977).

5. D.M. Ritchie and K. Thompson, "The UNIX Time-Sharing System," *The Bell System Technical Journal* **57**(6) pp. 1905-1931 (1978).

6. A. Jozwiak, T. Wollodko, L. Wisniewski, J. Rajtar, and J. Gomulka, "Production Model of Polish Agriculture," Technical report, International Institute for Applied Systems Analysis, Laxenburg, Austria (forthcoming).

7. L. Schrattenholzer, "The Energy Supply Model MESSAGE," RR-81-31, International Institute for Applied Systems Analysis, Laxenburg, Austria (1981).

8. S. Messner, "Users Guide for Message II," Technical report, International Institute for Applied Systems Analysis, Laxenburg, Austria (forthcoming).