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

Numerical Experiments with Decomposition of LP on a Small Computer

Nurminski, E.A.

IIASA Working Paper

WP-80-037

March 1980

NOT FOR QUOTATION
WITHOUT PERMISSION
OF THE AUTHOR

NUMERICAL EXPERIMENTS WITH DECOMPOSITION OF LP ON A SMALL COMPUTER

E. Nurminski

March 1980
WP-80-37

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

Numerical experiments with decomposition of LP on a small computer

E.Nurminski

IIASA

ABSTRACT

Results of numerical experiments with decomposition of linear programming problems are reported.

1. Introduction

There are many cases at IIASA where particular models developed in one project or program have natural connections with other models developed in different programs, areas or projects. The linkage of such models is of great interest and may lead to more comprehensive results. One of the hinderances on the way to integrating such models is the modest size of the IIASA computer- DEC PDP-11/70. As a rule every model developed at IIASA uses the computing power of the PDP-11/70 to its limit and some of them even beyond this limit. Naturally, it is very difficult to unite such models without substantial changes and this, to a great extent, prevents such linkage.

The purpose of this paper is to present some experimental results on the numerical effectivness of a decomposition approach. Decomposition of the original large-scale problem is in fact a specific way of linking subproblems or submodels to reach a coordinated solution. A substantial advantage of this approach is in a distributed manner of solving the original problem. In this case computations are performed with each subproblem separately, possibly even on different computers located at different institutions.

Experiments were conducted with LP but neither linearity nor the optimization nature of submodels are crucial for the approach.

2. Test problems

In an experimental application of the algorithm described in Nurminski (1979), randomly generated linear programming problems were solved using the DEC minicomputer PDP-11/70 under the UNIX (Ritche,Thompson,1978) operating system. These problems consist of two blocks with 39 rows and 100 colomns each and with a two-dimensional link between these blocks. These subproblems are referred to below as subproblems A and B respectively. Taken as a whole, this problem has 78 constraints and 198 variables, which is a rather modest size by today's standards. However, the PDP-11/70 has only 28K of core available to the user in regular mode under the UNIX time-sharing operating system, so it is unable to handle a problem of this size without a special means of operating system. It is worth noting that to store this matrix in the core in double precision one needs about 124K. Even if only non-zero elements of the original matrix are stored one still needs about 64K.

Coefficients of the constraint matrix and costs associated with variables were generated by the IMSL (IMSL 1977) subroutine *ggub* providng pseudo-random numbers uniformly distributed on [0, 1]. To avoid occasional degenereracy a special constraint on linking variables has been added

$$x_1 + x_2 \leq 1$$

to guarantee the boundness of the feasible set.

A Fortran text of the matrix generator is given in the Appendix. To avoid a trivial solution some elements of the constraint matrix and costs were made negative as shown in the Fortran text of the matrix generator.

Two problems of this kind were generated and solved. The initial integers (iseed) for generator *ggub* of random numbers were chosen as given in Table 1.

Table 1. Initial values for a random generator.

problem	a	b
test 1	5368	3568
test 2	23368	31057

3. Method

Generally speaking, the method applied for solving this problem consists of calculating some specific approximation of a coordinating function. This approximation provides enough information to define optimal values for linking variables leaving the decisions for choosing local variables to subproblems. The approximating function depends only on linking variables and a fairly simple structure of it can be made. For one of the test problems this function is shown in Fig. 1. The particular advantage of this function is that it has the same minimum as the original one and it reduces the initial problem to a problem of calculating this approximating function at a few test points. Actual calculation of the numerical value of approximation and its subgradient (this approximation is essentialy nondifferentiable) can be done in a decomposed way restricting computational efforts to those performed with subproblems separately. In this way rather large problems can be solved on small computers like the PDP 11/70.

An approximation of the coordinating function is to be calculated in a few basic points of the space of the linking variables. These points are further referred to as *reper* points to distinguish them from basis points of LP problems.

At every *reper* point it is necessary to solve a pair of master problem - subproblems of the kind which is typical for the Danzig-Wolfe decomposition scheme (Lasdon 1972). The master problem sends prices to the subproblems for linking variables and receives optimal values of linking variables and optimal values of objective functions in each of the subproblems. This process continues for some time until stopping criteria is satisfied. After completion of this cycle the value of approximation and its subgradient are used for computing new *reper* points or, if the number of these *reper* points is large enough, for computing the optimal solution. The essential difference with the Dantzig-Wolfe decomposition scheme is that local variables are controlled by subproblems exclusively. The optimal solution is reached by fixing optimal levels of linking variables rather then by directly prescribing optimal plans for subproblems.

4. Results

The full set of results is given in the Appendix; here we will discuss only some particular features of the method and its performance for given test problems.

In accordance with the theory pair "master-subproblems" should be solved for a set of *reper* points which may be chosen in a different way. Here we choose this set as follows:

r1= (0.0 , 0.0)
r2= (2.0 , 0.0)
r3= (2.0 , 2.0)

It is worth noting that points R2 and R3 are not feasible. Nevertheless, the method provides a finite value of approximation at these points as well as finite subgradients which show directions of possible changes in linking variables. Then a certain set of extreme prices was fixed for which the subproblems provided a bounded solution in a master problem. This set was chosen the same for all *reper* points and it consists of a few large price vectors. Generally, the master problem sends slightly different prices to the subproblems in the sense that their sum is not equal to zero.

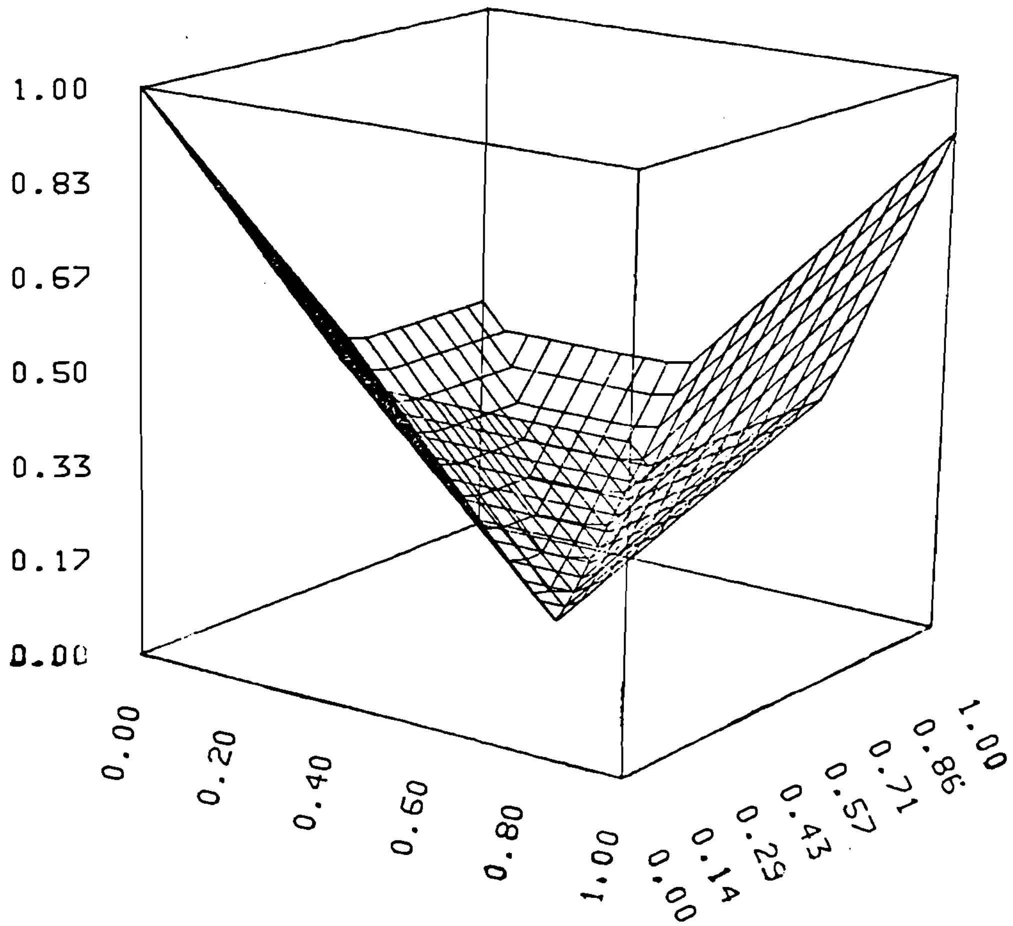

Figure 1. Approximating function.

During this initial phase this feature of the method was neglected and the sum of the prices was put equal to zero. So far as these initial results are necessary only to initiate the whole process, it does not influence the final results. An assumption about prices and results of computations in the subproblem are given in the following tables.

Table 2. Starting results for subproblem A.

test 1					test 2		
p1	p2	value	x1	x2	value	x1	x2
-1.0d 05	0.0d 00	-0.481d 00	0.1d 01	0.0d 01	-0.381d 00	0.1d 00	0.0d 00
0.0d 00	-1.0d 05	-0.861d 00	0.0d 00	0.1d 00	-0.467d 00	0.0d 00	0.1d 00

Table 3. Starting results for subproblem B.

test 1					test 2		
p1	p2	value	x1	x2	value	x1	x2
1.0d 05	0.0d 00	-1.090d 00	0.0d 00	0.1d 01	-0.612d 00	0.0d 00	0.1d 01
0.0d 00	1.0d 05	-0.404d 00	0.1d 01	0.0d 00	-0.943d 00	0.1d 01	0.0d 00

For a decomposition approach the number of cycles between master problem and subproblems is an important characteristic. For problems under consideration the following results were obtained.

Table 4. Number of cycles.

reper point	test1	test2
r1	6	6
r2	6	5
r3	7	4

This table shows that the number of cycles between master problem and subproblems are actually rather small. Moreover, so far as only cost coefficients of objective functions have been changed in local problems, the same solution of subproblems remained optimal in many cases.

For instance for test problem 1, at *reper* point r1 in subproblem A only 5 distinct solutions were generated and in subproblem B the actual number of different values for linking variables is 4.

After setting some initial framework for master problems, the process of trade-off between master problem and subproblems continued as is shown in detail in the Appendix. In the next two tables the final results for corresponding *reper* points are given.

Table 5. Test problem 1.

reper point	function	g(1)	g(2)
r1	-0.19791394d 01	0.0d 00	-0.1d 00
r2	-0.17929368d 01	0.1d 00	-0.1d 00
r3	-0.18791394d 01	0.1d 00	0.0d 00

where g(1) and g(2) are components of the subgradient of the approximating function with respect to linking variables. The next table has the same structure and relates to test problem 2.

Table 6. Test problem 2.

reper point	function	g(1)	g(2)
r1 (0,0)	-0.14092d 01	-0.1d 00	0.0d 00
r2 (2,0)	-0.12973d 01	0.1d 00	-0.1d 00
r3 (2,2)	-0.13092d 01	0.0d 00	0.1d 00

The final step of the method consists of solving a linear system of the size defined by the number of linking variables. In the case under consideration these systems have the following forms:

Test 1:

$$-0.19791394d\ 01 - 0.1\ x_2 = L$$

$$-0.17929368d\ 01 + 0.1\ (x_1 - 2) - 0.1\ x_2 = L$$

$$-0.18791394d\ 01 + 0.1\ (x_1 - 2.) = L$$

Test 2:

$$-0.14092d01 - 0.1\ x_1 = L$$

$$-0.12973d\ 01 + 0.1\ (x_1 - 2.) - 0.1\ x_2 = L$$

$$-0.13092d01 + 0.1\ (x_2 - 2.) = L$$

and their solutions are

Test 1:
x(1)= 0.013 x(2)= 0.987
Optimal value: -2.078

Test 2:
x(1)= 0.627 x(2)= 0.373
Optimal value: -1.472

5. Conclusions.

1 The decomposition approach provides an efficient algorithmic tool for solving large-scale problems. It consists of a separate consideration of submodels and offers a theoretical foundation for linkage procedures. In this approach local variables are treated locally and exchange is restriced to global variables.

2 Numerical experiments showed that the approach requires small information exchange between different subsystems and gives rapid convergency in coordinating process.

6. Appendix

6.1. Matrix generator

The matrix generator was written for an FTN compiler on the UNIX operating system.

```
c
c-genmat----d-----
c
c     function:  generate matrix for subproblem
c                and recode it in the file 1
c     usage:     call genmat(ma,na,nx)
c
c     parameters:      ma   -number of rows
c                      na   -number of columns
c                      nx   -number of lnks
c
c     subroutines:     ggub (IMSL library)
c
c     files:     1 write a   5d16.8
c
c-------------------------------------------
      subroutine genmat(ma,na,nx)
      dimension a (4000)
```

```
      double precision aa (4000)
      data iseed / 5368/
15    format(5d16.8)
25    format(3i4)
      call ggub(iseed,ma*na, a)
      do 10 ii= 1,ma*na
10    aa(ii)= a(ii)
c
c     slack part of matrix
c
      do 30 i= 1,ma**2
30    aa(i)= 0.
      do 20 i= 1,ma
20    aa(i+ (i-1)*ma)= 1.
c
c     negative prices
c
      do 40 i= ma+ 1,na-nx-1
      mai= ma*i
40    aa(mai)= aa(mai)*(-1)**i
      i0= (na-nx-1)*ma
c
      do 60 ii= 1,nx*ma
60    aa(i0+ ii)= -aa(i0+ ii)
c
c     binding x-links
c
      do 70 ii= 2,na
70    aa((ii-1)*ma+ 1)= 0.
      do 80 ii= 1,nx+ 1
80    aa(ma*(na-ii)+ 1)= 1.
c
c
      write (1,25) ma,na,nx
      write(1,15) (aa(ii),ii= 1,ma*na)
      return
      end
```

6.2. Output for the test problems.

This part presents all major information about details of computation for both test problems. It was obtained by combining outputs of subproblems and master problem in such a way that any information related to the computational process at some *reper* point can be looked through easily.

Test 1

for subproblem A iseed= 5368
for subproblem B iseed= 3568

reper point x= 0.0 0.0

master problem:

sequence of runs for master problem:

new run

```
 function value   0.13684301d 01
new run
 function value   0.19649653d 01
new run
 function value   0.19711405d 01
new run
            0.20140482d 01
 function value   0.19752512d 01
new run
            0.20140482d 01
            0.19822402d 01
 function value   0.19764435d 01
new run
            0.20140482d 01
            0.19822402d 01
 function value   0.19789357d 01
new run
            0.20140482d 01
            0.19822402d 01
            0.19792230d 01
 function value   0.19791136d 01  *********** final value at reper point
 subproblem A
      va          x(1)         x(2)
 -0.48095952d 00  0.10000000d 01  0.00000000d 00
 -0.86140775d 00  0.00000000d 00  0.10000000d 01
 -0.93308574d 00  0.12872318d 00  0.87127682d 00
 -0.86895934d 00  0.41497010d 00  0.58502990d 00
 -0.91142280d 00  0.28022948d 00  0.71977052d 00
 -0.92577755d 00  0.19241779d 00  0.80758221d 00
 -0.93308574d 00  0.12872318d 00  0.87127682d 00
 -0.93277678d 00  0.13258144d 00  0.86741856d 00
subproblem B
      va          x(1)         x(2)
 -0.10904710d 01  0.00000000d 00  0.10000000d 01
 -0.40402197d 00  0.10000000d 01  0.00000000d 00
 -0.11297777d 01  0.15609161d 00  0.84390839d 00
 -0.10913288d 01  0.18997125d-02  0.99810029d 00
 -0.11331024d 01  0.12753056d 00  0.87246944d 00
 -0.11331024d 01  0.12753056d 00  0.87246944d 00
 -0.11331072d 01  0.13904334d 00  0.86095666d 00
 -0.11331024d 01  0.12753056d 00  0.87246944d 00
prA
  0.00000000d 00  0.10041724d 00
prB
  0.00000000d 00 -0.41724000d-03

reper point x=  2.0 0.0

master problem:
      sequence of runs for master problem:
new run
            0.12684301d 01
 function value   0.12184301d 01
new run
```

```
 function value   0.17548417d 01
new run
            0.19190572d 01
 function value   0.17769298d 01
new run
            0.19190572d 01
            0.18161194d 01
            0.17888807d 01
 function value   0.17858024d 01
new run
            0.19190572d 01
            0.18161194d 01
 function value   0.17926578d 01
new run
            0.19190572d 01
            0.18161194d 01
            0.17943773d 01
 function value   0.17928175d 01
new run
            0.19190572d 01
            0.18161194d 01
            0.17943773d 01
 function value   0.17929368d 01  ********** final value at reper point
subproblem A
      va            x(1)           x(2)
 -0.48095952d 00  0.10000000d 01  0.00000000d 00
 -0.86140775d 00  0.00000000d 00  0.10000000d 01
 -0.93308574d 00  0.12872318d 00  0.87127682d 00
 -0.86895934d 00  0.41497010d 00  0.58502990d 00
 -0.91546184d 00  0.26407347d 00  0.73592653d 00
 -0.93308574d 00  0.12872318d 00  0.87127682d 00
 -0.92577755d 00  0.19241779d 00  0.80758221d 00
 -0.93277678d 00  0.13258144d 00  0.86741856d 00
subproblem B
      vb            x(1)           x(2)
 -0.10904710d 01  0.00000000d 00  0.10000000d 01
 -0.40402197d 00  0.10000000d 01  0.00000000d 00
 -0.11175160d 01  0.20460149d 00  0.79539851d 00
 -0.11273768d 01  0.10281919d 00  0.89718081d 00
 -0.11331072d 01  0.13904334d 00  0.86095666d 00
 -0.11331072d 01  0.13904334d 00  0.86095666d 00
 -0.11331072d 01  0.13904334d 00  0.86095666d 00
 -0.11331072d 01  0.13904334d 00  0.86095666d 00
prA
 -0.10000000d 00  0.16972750d-01
prB
  0.00000000d 00  0.83027250d-01

reper point x=  2.0 2.0

master problem:
      sequence of runs for master problem:
new run
 function value   0.12684301d 01
```

```
new run
 function value   0.18749014d 01
new run
 function value   0.18771815d 01
new run
 function value   0.18784902d 01
new run
                 0.18916476d 01
 function value   0.18788532d 01
new run
                 0.18916476d 01
                 0.18802174d 01
 function value   0.18789358d 01
new run
                 0.18916476d 01
                 0.18802174d 01
                 0.18792230d 01
                 0.18791373d 01
 function value   0.18791136d 01
new run
                 0.18916476d 01
                 0.18802174d 01
                 0.18792230d 01
 function value   0.18791394d 01  ************ final value at reper point
 subproblem A
      va            x(1)          x(2)
 -0.48095952d 00  0.10000000d 01  0.00000000d 00
 -0.86140775d 00  0.00000000d 00  0.10000000d 01
 -0.93308574d 00  0.12872318d 00  0.87127682d 00
 -0.86895934d 00  0.41497010d 00  0.58502990d 00
 -0.90532595d 00  0.30289483d 00  0.69710517d 00
 -0.91900322d 00  0.24362300d 00  0.75637700d 00
 -0.92577755d 00  0.19241779d 00  0.80758221d 00
 -0.93277678d 00  0.13258144d 00  0.86741856d 00
 -0.93277678d 00  0.13258144d 00  0.86741856d 00
subproblem B
      vb            x(1)          x(2)
 -0.10904710d 01  0.00000000d 00  0.10000000d 01
 -0.40402197d 00  0.10000000d 01  0.00000000d 00
 -0.11331072d 01  0.13904334d 00  0.86095666d 00
 -0.10913288d 01  0.18997125d-02  0.99810029d 00
 -0.11331024d 01  0.12753056d 00  0.87246944d 00
 -0.11331024d 01  0.12753056d 00  0.87246944d 00
 -0.11331024d 01  0.12753056d 00  0.87246944d 00
 -0.11331024d 01  0.12753056d 00  0.87246944d 00
 -0.11331072d 01  0.13904334d 00  0.86095666d 00
prA
 -0.10041698d 00  0.00000000d 00
prB
  0.41698321d-03  0.00000000d 00
SSSS
Test 2

for subproblem A iseed= 23368
```

for subproblem B iseed= 31057

reper point x= 2.0 0.0 eps= 0.1

master problem:

sequence of runs for master problem:

```
new run
 function value   0.96626960d 00
new run
 function value   0.12870340d 01
new run
             0.13284191d 01
             0.13282297d 01
 function value   0.12962132d 01
new run
             0.13565428d 01
             0.13373668d 01
             0.13126873d 01
             0.13105398d 01
             0.13067007d 01
 function value   0.12970694d 01
new run
             0.13565428d 01
             0.13373668d 01
             0.13126873d 01
             0.13105398d 01
             0.13067007d 01
 function value   0.12972768d 01
new run
             0.13565428d 01
             0.13475458d 01
             0.13403215d 01
             0.13242663d 01
             0.13126872d 01
             0.13105397d 01
             0.13067006d 01
             0.12977036d 01
 function value   0.12973135d 01     ******************** final value at reper point
```

subproblem A

va	x(1)	x(2)
-0.31085498d 00	0.10000000d 01	0.00000000d 00
-0.46719693d 00	0.00000000d 00	0.10000000d 01
-0.45362591d 00	0.72178433d 00	0.27821567d 00
-0.48793737d 00	0.50157846d 00	0.49842154d 00
-0.48793737d 00	0.50157846d 00	0.49842154d 00
-0.45362591d 00	0.72178433d 00	0.27821567d 00
-0.48793737d 00	0.50157846d 00	0.49842154d 00

subproblem B

vb	x(1)	x(2)
-0.61186347d 00	0.00000000d 00	0.10000000d 01
-0.94262375d 00	0.10000000d 01	0.00000000d 00
-0.10023808d 01	0.64656685d 00	0.35343315d 00
-0.990032386d 00	0.82120677d 00	0.17879323d 00
-0.99927155d 00	0.71951552d 00	0.28048448d 00

```
 -0.10022263d 01  0.65625333d 00  0.34374667d 00
 -0.10022263d 01  0.65625333d 00  0.34374667d 00
prA
 -0.15581535d 00  0.00000000d 00
prB
  0.55815352d-01  0.10000000d 00

reper point x=  0.0 0.0   eps=  0.1
master problem:
      sequence of runs for master problem:
new run
 function value   0.11162696d 01
new run
 function value   0.13856466d 01
new run
 function value   0.13954441d 01
new run
 function value   0.14030701d 01
new run
 function value   0.14092139d 01
new run
              0.14092412d 01
 function value   0.14092139d 01   *************** final value at reper point
subproblem A
      va           x(1)          x(2)
 -0.31085498d 00  0.10000000d 01  0.00000000d 00
 -0.46719693d 00  0.00000000d 00  0.10000000d 01
 -0.45362591d 00  0.72178433d 00  0.27821567d 00
 -0.48535826d 00  0.13878891d 00  0.86121109d 00
 -0.48793737d 00  0.50157846d 00  0.49842154d 00
 -0.48793737d 00  0.50157846d 00  0.49842154d 00
 -0.48793737d 00  0.50157846d 00  0.49842154d 00
 -0.48793737d 00  0.50157846d 00  0.49842154d 00
subproblem B
      vb           x(1)          x(2)
 -0.61186347d 00  0.00000000d 00  0.10000000d 01
 -0.94262375d 00  0.10000000d 01  0.00000000d 00
 -0.10023808d 01  0.64656685d 00  0.35343315d 00
 -0.10023808d 01  0.64656685d 00  0.35343315d 00
 -0.10023808d 01  0.64656685d 00  0.35343315d 00
 -0.96975616d 00  0.49371600d 00  0.50628400d 00
 -0.96975616d 00  0.49371600d 00  0.50628400d 00
 -0.10004230d 01  0.63738617d 00  0.36261383d 00
prA
  0.00000000d 00  0.11345308d 00
prB
  0.10000000d 00 -0.11345308d 00

reper point x=  2.0 2.0  eps=  0.1
master problem:
      sequence of runs for master problem:
new run
 function value   0.12620896d 01
new run
```

```
           0.12974379d 01
 function value   0.12892198d 01
new run
           0.13266922d 01
 function value   0.13055873d 01
new run
           0.13266922d 01
 function value   0.13092138d 01   **************** final value at reper point
subproblem  A
     va          x(1)        x(2)
 -0.31085498d 00  0.10000000d 01  0.00000000d 00
 -0.467196 93d 00  0.00000000d 00  0.10000000d 01
 -0.45362591d 00  0.72178433d 00  0.27821567d 00
 -0.48535826d 00  0.13878891d 00  0.86121109d 00
 -0.487937 37d 00  0.50157846d 00  0.49842154d 00
 -0.487937 37d 00  0.50157846d 00  0.49842154d 00
subproblem  B
     vb          x(1)        x(2)
 -0.61186347d 00  0.00000000d 00  0.10000000d 01
 -0.94262375d 00  0.10000000d 01  0.00000000d 00
 -0.969756 16d 00  0.49371600d 00  0.50628400d 00
 -0.10023808d 01  0.64656685d 00  0.35343315d 00
 -0.10023808d 01  0.64656685d 00  0.35343315d 00
 -0.969756 16d 00  0.49371600d 00  0.50628400d 00
prA
 -0.21345307d 00 -0.10000000d 00
prB
  0.21345307d 00  0.00000000d 00

optimal point x=  0.627 0.373
master problem:
     sequence of runs for master problem:
new run
 function value   0.11408696d 01
new run
 function value   0.14439175d 01
new run
           0.14581441d 01
 function value   0.14561874d 01
new run
 function value   0.14598474d 01
new run
 function value   0.14665991d 01
new run
           0.14666028d 01
 function value   0.14665997d 01
new run
           0.14666028d 01
 function value   0.14665999d 01
new run
           0.14666028d 01
 function value   0.14666007d 01 *********** final value at the optimal point

 subproblem A
```

va	x(1)	x(2)
-0.31085498d 00	0.10000000d 01	0.00000000d 00
-0.46719693d 00	0.00000000d 00	0.10000000d 01
-0.45362591d 00	0.72178433d 00	0.27821567d 00
-0.48535826d 00	0.13878891d 00	0.86121109d 00
-0.48793737d 00	0.50157846d 00	0.49842154d 00
-0.48793737d 00	0.50157846d 00	0.49842154d 00
-0.48793737d 00	0.50157846d 00	0.49842154d 00
-0.45362591d 00	0.72178433d 00	0.27821567d 00
-0.45362591d 00	0.72178433d 00	0.27821567d 00

subproblem B

vb	x(1)	x(2)
-0.61186347d 00	0.00000000d 00	0.10000000d 01
-0.94262375d 00	0.10000000d 01	0.00000000d 00
-0.10023808d 01	0.64656685d 00	0.35343315d 00
-0.10023808d 01	0.64656685d 00	0.35343315d 00
-0.969975616d 00	0.49371600d 00	0.50628400d 00
-0.10023808d 01	0.64656685d 00	0.35343315d 00
-0.10004230d 01	0.63738617d 00	0.36261383d 00
-0.969975616d 00	0.49371600d 00	0.50628400d 00
-0.10004230d 01	0.63738617d 00	0.36261383d 00

prA
0.00000000d 00 0.15581537d 00
prB
0.57637704d-01 -0.15581537d 00

REFERENCES

Lasdon, Leon S. 1972. Optimization Theory for Large Systems. New York: The MacMillan Company.

Nurminski, E. 1979. Some Theoretical Considerations on Linkage Problems. WP-79-117. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Ritchie, D.M., and K. Thompson. 1978. The UNIX time-sharing system. The Bell System Technical Journal, July-August 1979, 57(6), Part 2:1905-1930.