BENDERS DECOMPOSITION WITH GAMS

ERWIN KALVELAGEN

Abstract. This document describes an implementation of Benders Decomposition using GAMS.

1. Introduction

Benders' Decomposition[1] is a popular technique in solving certain classes of difficult problems such as stochastic programming problems[4, 7] and mixed-integer nonlinear programming problems[3, 2]. In this document we describe how a Benders' Decomposition algorithm can be implemented in a GAMS environment.

2. Benders' Decomposition for MIP Problems

Using the notation in [6] , we can state the MIP problem as:

$$\text{MIP} \qquad \begin{aligned} \underset{x,y}{\text{minimize}} \quad & c^T x + f^T y \\ & Ax + By \geq b \\ & y \in Y \\ & x \geq 0 \end{aligned}$$

If y is fixed to a feasible integer configuration, the resulting model to solve is:

$$\begin{aligned} \min_x \quad & c^T x \\ & Ax \geq b - B\overline{y} \\ & x \geq 0 \end{aligned} \tag{1}$$

The complete minimization problem can therefore be written as:

$$\min_{y \in Y} \left[f^T y + \min_{x \geq 0} \{ c^T x | Ax \geq b - By \} \right] \tag{2}$$

The dual of the inner LP problem is:

$$\begin{aligned} \max_u \quad & (b - B\overline{y})^T u \\ & A^T u \leq c \\ & u \geq 0 \end{aligned} \tag{3}$$

The Benders' Decomposition algorithm can be stated as:

{initialization}

$y :=$ initial feasible integer solution
$LB := -\infty$
$UB := \infty$

Date: 20 december 2002.

while $UB - LB > \epsilon$ **do**
  {solve subproblem}
  $\min_u\{f^T\overline{y} + (b - B\overline{y})^T u | A^T u \le c, u \ge 0\}$
  if Unbounded **then**
    Get unbounded ray $\overline{u}$
    Add cut $(b - By)^T\overline{u} \le 0$ to master problem
  else
    Get extreme point $\overline{u}$
    Add cut $z \ge f^T y + (b - By)^T\overline{u}$ to master problem
    $UB := \min\{UB, f^T\overline{y} + (b - B\overline{y})^T\overline{u}\}$
  end if
  {solve master problem}
  $\min_y\{z|\text{cuts}, y \in Y\}$
  $LB := \overline{z}$
end while

The subproblem is a dual LP problem, and the master problem is a pure IP problem (no continuous variables are involved). Benders' Decomposition for MIP is of special interest when the Benders' subproblem and the relaxed master problem are easy to solve, while the original problem is not.

3. The Fixed Charge Transportation Problem

The problem we consider is the Fixed Charge Transportation Problem (FCTP). The standard transportation problem can be described as:

TP

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{i,j} c_{i,j} x_{i,j} \\ & \sum_j x_{i,j} = s_i \\ & \sum_i x_{i,j} = d_j \\ & x_{i,j} \ge 0 \end{aligned}$$

The fixed charge transportation problem adds a fixed cost $f_{i,j}$ to a link $i \to j$. This can be modeled using extra binary variables $y_{i,j}$ indicating whether a link is open or closed:

FCTP

$$\begin{aligned} \underset{x,y}{\text{minimize}} \quad & \sum_{i,j} (f_{i,j} y_{i,j} + c_{i,j} x_{i,j}) \\ & \sum_j x_{i,j} = s_i \\ & \sum_i x_{i,j} = d_j \\ & x_{i,j} \le M_{i,j} y_{i,j} \\ & x_{i,j} \ge 0, y_{i,j} \in \{0,1\} \end{aligned}$$

where $M_{i,j}$ are large enough numbers. When solving this as a straight MIP problem, it is important to assign reasonable values to $M_{i,j}$. As $M_{i,j}$ can be considered as an upper bound on $x_{i,j}$, we can find good values:

$$M_{i,j} = \min\{s_i, d_j\} \tag{4}$$

When we rewrite the problem as

$$\begin{aligned}
&\min_{x,y} \sum_{i,j} c_{i,j}x_{i,j} + \sum_{i,j} f_{i,j}y_{i,j} \\
&\quad -\sum_j x_{i,j} \ge -s_i \\
&\quad \sum_i x_{i,j} \ge d_j \\
&\quad -x_{i,j} + M_{i,j}y_{i,j} \ge 0 \\
&\quad x_{i,j} \ge 0 \\
&\quad y_{i,j} \in \{0,1\}
\end{aligned} \tag{5}$$

we see that the Benders' subproblem can be stated as:

$$\begin{aligned}
&\max_{u,v,w} \sum_i (-s_i)u_i + \sum_j d_j v_j + \sum_{i,j}(-M_{i,j}\overline{y}_{i,j})w_{i,j} \\
&\quad -u_i + v_j - w_{i,j} \le c_{i,j} \\
&\quad u_i \ge 0, v_j \ge 0, w_{i,j} \ge 0
\end{aligned} \tag{6}$$

The Benders' Relaxed Master Problem can be written as:

$$\begin{aligned}
&\min_y z \\
&\quad z \ge \sum_{i,j} f_{i,j}y_{i,j} + \sum_i (-s_i)\overline{u}_i^{(k)} + \sum_j d_j \overline{v}_j^{(k)} + \sum_{i,j}(-M_{i,j}\overline{w}_{i,j}^{(k)})y_{i,j} \\
&\quad \sum_i (-s_i)\overline{u}_i^{(\ell)} + \sum_j d_j \overline{v}_j^{(\ell)} + \sum_{i,j}(-M_{i,j}\overline{w}_{i,j}^{(\ell)})y_{i,j} \le 0 \\
&\quad y_{i,j} \in \{0,1\}
\end{aligned} \tag{7}$$

Using this result the GAMS model can now be formulated as:

Model benders.gms. [1]

```
$ontext

   An example of Benders Decomposition on fixed charge transportation
   problem bk4x3.

   Optimal objective in reference : 350.

   Erwin Kalvelagen, December 2002

   See:
   http://www.in.tu-clausthal.de/~gottlieb/benchmarks/fctp/

$offtext


set i 'sources' /i1*i4/;
set j 'demands' /j1*j3/;

parameter supply(i) /
   i1 10
   i2 30
```

[1] http://www.gams.com/~erwin/benders/benders.gms

```
   i3 40
   i4 20
/;


parameter demand(j) /
   j1 20
   j2 50
   j3 30
/;


table c(i,j) 'variable cost'
       j1    j2    j3
i1    2.0   3.0   4.0
i2    3.0   2.0   1.0
i3    1.0   4.0   3.0
i4    4.0   5.0   2.0
;



table f(i,j) 'fixed cost'
        j1     j2     j3
i1    10.0   30.0   20.0
i2    10.0   30.0   20.0
i3    10.0   30.0   20.0
i4    10.0   30.0   20.0
;

*
* check supply-demand balance
*
scalar totdemand, totsupply;
totdemand = sum(j, demand(j));
totsupply = sum(i, supply(i));
abort$(abs(totdemand-totsupply)>0.001) "Supply does not equal demand.";

*
* for big-M formulation we need tightest possible upperbounds on x
*
parameter xup(i,j) 'tight upperbounds for x(i,j)';
xup(i,j) = min(supply(i),demand(j));


*------------------------------------------------------------------
* standard MIP problem formulation
*------------------------------------------------------------------

variables
   cost   'objective variable'
   x(i,j) 'shipments'
   y(i,j) 'on-off indicator for link'
;
positive variable x;
binary variable y;

equations
   obj      'objective'
   cap(i)   'capacity constraint'
   dem(j)   'demand equation'
   xy(i,j)  'y=0 => x=0'
;

obj..        cost =e= sum((i,j), f(i,j)*y(i,j) + c(i,j)*x(i,j));
cap(i)..     sum(j, x(i,j)) =l= supply(i);
dem(j)..     sum(i, x(i,j)) =g= demand(j);
xy(i,j)..    x(i,j) =l= xup(i,j)*y(i,j);

option optcr=0;
model fscp /obj,cap,dem,xy/;
solve fscp minimizing cost using mip;
```

```
*---------------------------------------------------------------------
* Benders Decomposition Initialization
*---------------------------------------------------------------------

scalar UB 'upperbound' /INF/;
scalar LB 'lowerbound' /-INF/;

y.l(i,j) = 1;


*---------------------------------------------------------------------
* Benders Subproblem
*---------------------------------------------------------------------

variable z 'objective variable';

positive variables
   u(i) 'duals for capacity constraint'
   v(j) 'duals for demand constraint'
   w(i,j) 'duals for xy constraint'
;

equations
   subobj          'objective'
   subconstr(i,j)  'dual constraint'
;

* to detect unbounded subproblem
scalar unbounded /1.0e6/;
z.up = unbounded;

subobj..  z =e= sum(i, -supply(i)*u(i)) + sum(j, demand(j)*v(j))
                + sum((i,j), -xup(i,j)*y.l(i,j)*w(i,j))
                 ;

subconstr(i,j)..  -u(i) + v(j) - w(i,j) =l= c(i,j);

model subproblem /subobj, subconstr/;

*---------------------------------------------------------------------
* Benders Modified Subproblem to find unbounded ray
*---------------------------------------------------------------------

variable dummy 'dummy objective variable';

equations
   modifiedsubobj          'objective'
   modifiedsubconstr(i,j)  'dual constraint'
   edummy;
;

modifiedsubobj..
    sum(i, -supply(i)*u(i)) + sum(j, demand(j)*v(j))
           + sum((i,j), -xup(i,j)*y.l(i,j)*w(i,j)) =e= 1;

modifiedsubconstr(i,j)..
    -u(i) + v(j) - w(i,j) =l= 0;

edummy.. dummy =e= 0;

model modifiedsubproblem /modifiedsubobj, modifiedsubconstr, edummy/;


*---------------------------------------------------------------------
* Benders Relaxed Master Problem
*---------------------------------------------------------------------

set iter /iter1*iter50/;

set cutset(iter) 'dynamic set';
cutset(iter)=no;
```

```
set unbcutset(iter) 'dynamic set';
unbcutset(iter)=no;


variable z0 'relaxed master objective variable';
equations
   cut(iter)           'Benders cut for optimal subproblem'
   unboundedcut(iter)  'Benders cut for unbounded subproblem'
;

parameters
   cutconst(iter)     'constant term in cuts'
   cutcoeff(iter,i,j)
;

cut(cutset).. z0 =g= sum((i,j), f(i,j)*y(i,j))
                      + cutconst(cutset)
                      + sum((i,j), cutcoeff(cutset,i,j)*y(i,j));
unboundedcut(unbcutset)..
                cutconst(unbcutset)
                + sum((i,j), cutcoeff(unbcutset,i,j)*y(i,j)) =l= 0;



model master /cut,unboundedcut/;


*----------------------------------------------------------------------
* Benders Algorithm
*----------------------------------------------------------------------


loop(iter,

*
* solve Benders subproblem
*
   solve subproblem maximizing z using lp;

*
* check results.
*

   abort$(subproblem.modelstat>=2) "Subproblem not solved to optimality";

*
* was subproblem unbounded?
*

   if (z.l+1 < unbounded,

*
* no, so update upperbound
*

      UB = min(UB, sum((i,j), f(i,j)*y.l(i,j)) + z.l);

*
* and add Benders' cut to Relaxed Master
*
      cutset(iter) = yes;

   else

*
* solve modified subproblem
*

     solve modifiedsubproblem maximizing dummy using lp;

*
* check results.
```

```
*

     abort$(modifiedsubproblem.modelstat>=2)
            "Modified subproblem not solved to optimality";


*
* and add Benders' cut to Relaxed Master
*
      unbcutset(iter) = yes;
   );


*
* cut data
*
   cutconst(iter) = sum(i, -supply(i)*u.l(i)) + sum(j, demand(j)*v.l(j));
   cutcoeff(iter,i,j) = -xup(i,j)*w.l(i,j);

*
* solve Relaxed Master Problem
*

   option optcr=0;
   solve master minimizing z0 using mip;

*
* check results.
*

   abort$(master.modelstat=4) "Relaxed Master is infeasible";
   abort$(master.modelstat>=2) "Masterproblem not solved to optimality";

*
* update lowerbound
*

   LB = z0.l;

   display UB,LB;
   abort$( (UB-LB) < 0.1 ) "Converged";

);
```

The Benders' algorithm will converge to the optimal solution in 11 cycles. The values of the bounds are as follows:

cycle	LB	UB
1	250	460
2	260	460
3	280	460
4	310	460
5	320	460
6	330	460
7	330	460
8	340	410
9	340	410
10	340	410
11	350	350

4. Conclusion

We have shown how a standard Benders' Decomposition algorithm can be implemented in GAMS. Algorithmic development using a high level modeling language like GAMS is particular useful if complex subproblems need to be solved that can take advantage of the direct availability of the state-of-the-art LP, MIP or NLP capabilities of GAMS. Another example of such an exercise is found in [5] where a special form of a Generalized Benders' Decomposition is used to solve a MINLP problem. A related use of GAMS is as a prototyping language. In this case a GAMS implementation of an algorithm is used to test the feasibility and usefulness of a certain computational approach. In a later stage the algorithm can be formalized and implemented in a more traditional language. Indeed, this is the way solvers like SBB and DICOPT have been developed.

References

1. J. F. Benders, *Partitioning procedures for solving mixed-variables programming problems*, Numerische Mathematik **4** (1962), 238–252.
2. C. A. Floudas, A. Aggarwal, and A. R. Ciric, *Global optimum search for nonconvex NLP and MINLP problems*, Computers and Chemical Engineering **13** (1989), no. 10, 1117–1132.
3. A. M. Geoffrion, *Generalized Benders decomposition*, Journal of Optimization Theory and Applications **10** (1972), no. 4, 237–260.
4. Gerd Infanger, *Planning under uncertainty – solving large-scale stochastic linear programs*, Boyd & Fraser, 1994.
5. G. E. Paules IV and C. A. Floudas, *APROS: Algorithmic development methodology for discrete-continuous optimization problems*, Operations Research **37** (1989), no. 6, 902–915.
6. Richard Kipp Martin, *Large scale linear and integer optimization; a unified approach*, Kluwer, 1999.
7. S. S. Nielsen and S.A. Zenios, *Scalable parallel benders decomposition for stochastic linear programming*, Parallel Computing **23** (1997), 1069–1088.

GAMS Development Corp., Washington D.C.
E-mail address: erwin@gams.com