BENDERS DECOMPOSITION WITH GAMS

ERWIN KALVELAGEN

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

1. Introduction

Benders’ Decomposition\cite{1} is a popular technique in solving certain classes of dif-
ficult problems such as stochastic programming problems\cite{4, 7} and mixed-integer nonlinear programming problems\cite{9, 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 \cite{6}, we can state the MIP problem as:

\[
\text{MIP} \quad \min_{x,y} \quad c^T x + f^T y \\
\text{s.t.} \quad Ax + By \geq b \\
\quad y \in Y \\
\quad x \geq 0
\]

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

\[
\min_{x} \quad c^T x \\
\text{s.t.} \quad Ax \geq b - B\bar{y} \\
\quad x \geq 0
\]

(1)

The complete minimization problem can therefore be written as:

\[
\min_{y \in Y} \left[ f^T y + \min_{x \geq 0} \{ c^T x | Ax \geq b - B\bar{y} \} \right]
\]

(2)

The dual of the inner LP problem is:

\[
\max_{u} \quad (b - B\bar{y})^T u \\
\text{s.t.} \quad A^T u \leq c \\
\quad u \geq 0
\]

(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}

$\min_u \{ f^T \bar{y} + (b - B \bar{y})^T u | A^T u \leq c, u \geq 0 \}$

if Unbounded then

Get unbounded ray $\bar{n}$
Add cut $(b - By)^T n \leq 0$ to master problem
else

Get extreme point $\bar{n}$
Add cut $z \geq f^T y + (b - By)^T n$ to master problem
$UB := \min \{ UB, f^T \bar{y} + (b - B \bar{y})^T \bar{n} \}$

end if

{solve master problem}

$\min y \{ z | \text{cuts}, y \in Y \}$

$LB := z$

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:

\[
\begin{align*}
\text{TP} & \quad \text{minimize} \sum_{i,j} c_{i,j} x_{i,j} \\
& \quad \sum_{j} x_{i,j} = s_{i} \\
& \quad \sum_{i} x_{i,j} = d_{j} \\
& \quad x_{i,j} \geq 0
\end{align*}
\]

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

\[
\begin{align*}
\text{FCTP} & \quad \text{minimize} \sum_{i,j} (f_{i,j} y_{i,j} + c_{i,j} x_{i,j}) \\
& \quad \sum_{j} x_{i,j} = s_{i} \\
& \quad \sum_{i} x_{i,j} = d_{j} \\
& \quad x_{i,j} \leq M_{i,j} y_{i,j} \\
& \quad x_{i,j} \geq 0, y_{i,j} \in \{0, 1\}
\end{align*}
\]

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:

(4) \[ M_{i,j} = \min\{s_i, d_j\} \]

When we rewrite the problem as

\[
\begin{align*}
\min_{x,y} & \sum_{i,j} c_{i,j} x_{i,j} + \sum_{i,j} f_{i,j} y_{i,j} \\
- & \sum_{j} x_{i,j} \geq -s_i \\
\sum_{i} x_{i,j} & \geq d_j \\
- & x_{i,j} + M_{i,j} y_{i,j} \geq 0 \\
x_{i,j} & \geq 0 \\
y_{i,j} & \in \{0,1\}
\end{align*}
\]

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

(5) \[ \begin{align*}
\max_{u,v,w} & \sum_{i} (-s_i) u_i + \sum_{j} d_j v_j + \sum_{i,j} (-M_{i,j} y_{i,j}) w_{i,j} \\
- & u_i + v_j - w_{i,j} \leq c_{i,j} \\
u_i \geq 0, v_j \geq 0, w_{i,j} & \geq 0
\end{align*} \]

The Benders' Relaxed Master Problem can be written as:

(6) \[ \begin{align*}
\min y & \sum_{i,j} f_{i,j} y_{i,j} + \sum_{i} (-s_i) \mu_i^{(k)} + \sum_{j} d_j \nu_j^{(k)} + \sum_{i,j} (-M_{i,j} \tau_{i,j}^{(k)}) y_{i,j} \\
\sum_{i} (-s_i) \mu_i^{(l)} + \sum_{j} d_j \nu_j^{(l)} + \sum_{i,j} (-M_{i,j} \tau_{i,j}^{(l)}) y_{i,j} & \leq 0 \\
y_{i,j} & \in \{0,1\}
\end{align*} \]

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

Model benders.gms. [1]

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/

set i 'sources' /i1*i4/;
set j 'demands' /j1*j3/;
parameter supply(i) /
  i1 10
  i2 30

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

<table>
<thead>
<tr>
<th>cycle</th>
<th>LB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>250</td>
<td>460</td>
</tr>
<tr>
<td>2</td>
<td>260</td>
<td>460</td>
</tr>
<tr>
<td>3</td>
<td>280</td>
<td>460</td>
</tr>
<tr>
<td>4</td>
<td>310</td>
<td>460</td>
</tr>
<tr>
<td>5</td>
<td>320</td>
<td>460</td>
</tr>
<tr>
<td>6</td>
<td>330</td>
<td>460</td>
</tr>
<tr>
<td>7</td>
<td>330</td>
<td>460</td>
</tr>
<tr>
<td>8</td>
<td>340</td>
<td>410</td>
</tr>
<tr>
<td>9</td>
<td>340</td>
<td>410</td>
</tr>
<tr>
<td>10</td>
<td>340</td>
<td>410</td>
</tr>
<tr>
<td>11</td>
<td>350</td>
<td>350</td>
</tr>
</tbody>
</table>

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


GAMS Development Corp., Washington D.C.

E-mail address: erwin@gams.com