Benders' Decomposition Algorithm

This Hypercard stack was prepared by:
Dennis L. Bricker,
Dept. of Industrial Engineering,
University of Iowa,
Iowa City, Iowa 52242
e-mail: dbricker@icaen.uiowa.edu

author

Benders' Decomposition

(also known as Benders' Partitioning)

- Theory
- Applications
  - Capacitated Plant Location
  - Stochastic LP with Recourse

Consider the problem

$$\begin{array}{ll} \text{Minimize} & cx + dy \\ \text{subject to} & Ax + By \geq b \\ & x \geq 0 \\ & y \in Y \end{array}$$

The variables x are continuous, but the variables y are "complicating" in some way...
often

$$Y = \{ y \mid y_i \in \{0,1\} \}$$

i.e., y is binary integer.

A key concept in Benders' algorithm is that of partitioning the variables into two sets (x & y) and "projecting" the problem onto the y variables.

Define $v(y) = dy + \min \{ cx \mid Ax \geq b - By,\ x \geq 0 \}$

The original problem is clearly seen to be equivalent to:

$$\begin{array}{ll} \text{Minimize} & v(y) \\ \text{subject to} & y \in Y \end{array}$$

Evaluating v(y) entails solving an LP problem in x, or, by LP duality theory, its dual LP:

$$v(y) = dy + \max \{ (b-By)^T u \mid A^T u \leq c,\ u \geq 0 \}$$

What are the characteristics of this function?

For simplicity, assume that the primal LP

$$\min \{ cx \mid Ax \geq b - By,\ x \geq 0 \}$$

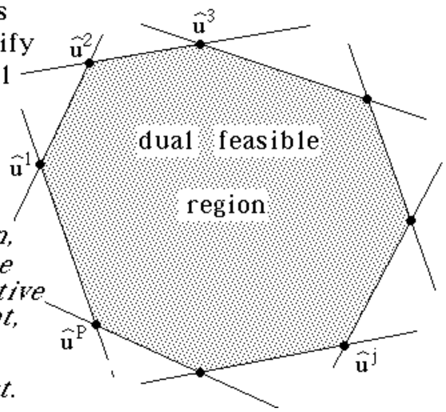
is always feasible for every choice of Y (e.g., x includes "artificial" variables with high costs).

Then the dual LP

$$\max \{ (b-By)^T u \mid A^T u \leq c,\ u \geq 0 \}$$

has a bounded feasible region.

In principle, it is possible to identify and enumerate all of the extreme points of the dual feasible region.



In principle, then, one could evaluate the dual LP objective at each extreme pt,

$$(b-By)^T \hat{u}^j$$

& choose the best.

That is, we can evaluate the function v(y) by

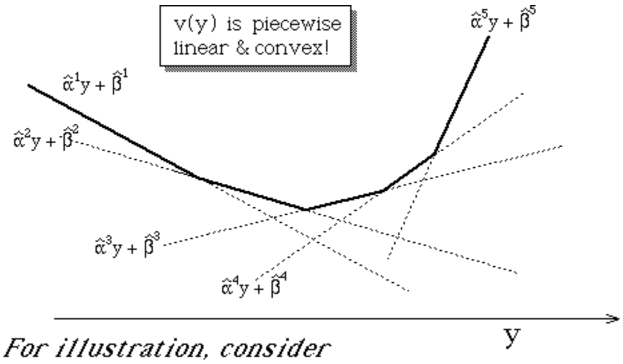
$$v(y) = dy + \underset{1 \leq j \leq P}{\text{maximum}} \{ (b-By)^T \hat{u}^j \}$$

or

$$v(y) = \underset{1 \leq j \leq P}{\text{maximum}} \{ \hat{\alpha}^j y + \hat{\beta}^j \}$$

where $\hat{\alpha}^j = [\hat{u}^j]^T B + d$, $\hat{\beta}^j = b^T \hat{u}^j$

So we see that the function v(y) is the maximum of a (large) set of linear functions in y!



For illustration, consider
Y = real numbers.

EXAMPLE

$$
\begin{aligned}
\text{Minimize}\quad & 18x_1 + 8x_2 + 20x_3 + 8y \\
\text{subject to}\quad & 3x_1 + x_2 + x_3 + 2y \ge 6 \\
& x_1 + x_2 + 4x_3 + y \ge 10 \\
& x_j \ge 0,\ j=1,2,3,4 \\
& y \in \{0, 1, 2, 3, \ldots 12\}
\end{aligned}
$$

Define

$$
v(y) = 8y + \left\{
\begin{aligned}
& \min 18x_1 + 8x_2 + 20x_3 \\
& \text{subject to}\quad 3x_1 + x_2 + x_3 \ge 6 - 2y \\
& \qquad x_1 + x_2 + 4x_3 \ge 10 - y \\
& \qquad x_j \ge 0,\ j=1,2,3,4
\end{aligned}
\right.
$$

The function v may be evaluated by solving either

Primal LP

$$
v(y) = 8y + \left\{
\begin{aligned}
& \min 18x_1 + 8x_2 + 20x_3 \\
& \text{subject to}\quad 3x_1 + x_2 + x_3 \ge 6 - 2y \\
& \qquad x_1 + x_2 + 4x_3 \ge 10 - y \\
& \qquad x_j \ge 0,\ j=1,2,3,4
\end{aligned}
\right.
$$

or

Dual LP

$$
v(y) = 8y + \left\{
\begin{aligned}
& \max (6-2y)u_1 + (10-y)u_2 \\
& \text{subject to}\quad 3u_1 + u_2 \le 18 \\
& \qquad u_1 + u_2 \le 8 \\
& \qquad u_1 + 4u_2 \le 20 \\
& \qquad u_1 \ge 0,\ u_2 \ge 0
\end{aligned}
\right.
$$

Note

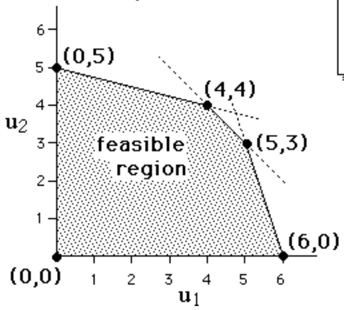
maximization is with respect to u_1 and u_2 with y temporarily fixed

$$
v(y) = 8y + \left\{
\begin{aligned}
& \max (6-2y)u_1 + (10-y)u_2 \\
& \text{subject to}\quad 3u_1 + u_2 \le 18 \\
& \qquad u_1 + u_2 \le 8 \\
& \qquad u_1 + 4u_2 \le 20 \\
& \qquad u_1 \ge 0,\ u_2 \ge 0
\end{aligned}
\right.
$$

The dual feasible region doesn't depend upon the value of y

Dual LP

The dual feasible region has five extreme points



$$
\begin{aligned}
& \max (6-2y)u_1 + (10-y)u_2 \\
& \text{subject to}\quad 3u_1 + u_2 \le 18 \\
& \qquad u_1 + u_2 \le 8 \\
& \qquad u_1 + 4u_2 \le 20 \\
& \qquad u_1 \ge 0,\ u_2 \ge 0
\end{aligned}
$$

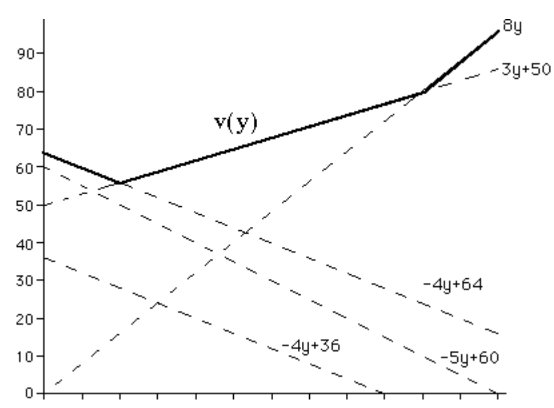
The solution of the LP must be one of these extreme points.

Extreme point $\hat{u}$	$8y + (6-2y)u_1 + (10-y)u_2$
(0,0)	8y
(0,5)	3y + 50
(4,4)	−4y + 64
(5,3)	−5y + 60
(6,0)	−4y + 36

For any y, the value of v(y) is the maximum of these five linear functions

EXAMPLE

Support	Y = 1	3	5	7	9
8y	8	24	40	56	72
3y + 50	53	59	65	71	77
−4y + 64	60	52	44	36	28
−5y + 60	55	45	35	25	15
−4y + 36	32	24	16	8	0



Note, however, that v(y) is to be evaluated by solving a linear programming problem, not by identifying all of the dual extreme points and computing the corresponding linear function of y.

The number of linear functions which define v(y) is, in general, "astronomical" !

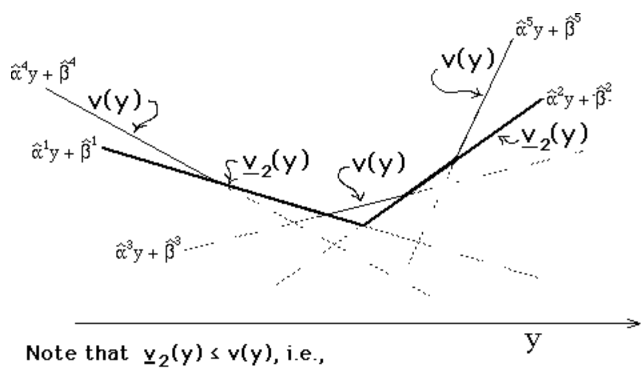
Approximating the function v(y)

Suppose that v(y) is the maximum of P linear functions ("supports")

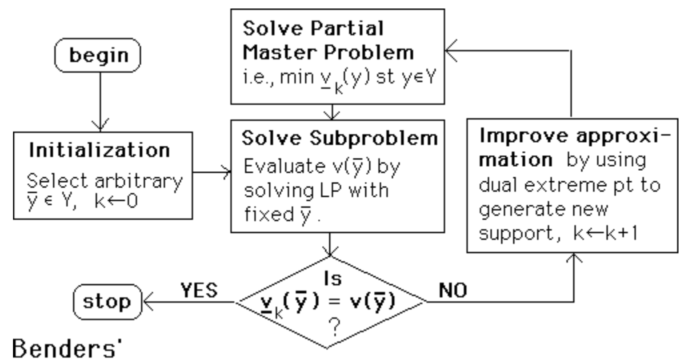
$$v(y) = \underset{1\le j\le P}{\text{maximum}}\left\{\hat{\alpha}^j y + \hat{\beta}^j\right\}$$

If k supports are used (where k<P), we get an *underestimate* of v(y):

$$\boxed{\underline{v}_k(y) = \underset{1\le j\le k}{\text{maximum}}\left\{\hat{\alpha}^j y + \hat{\beta}^j\right\}}$$



Note that $\underline{v}_2(y) \le v(y)$, **i.e., it underestimates v(y)**



Benders' Decomposition Algorithm

Solving the Partial Master Problem

At each iteration, we must solve

$$\boxed{\underset{y\in Y}{\text{Minimize}}\ \underline{v}_k(y)}$$

where $\underline{v}_k(y)$ is the current approximation to v(y),

that is, $\boxed{\underline{v}_k(y) = \underset{1\le j\le k}{\text{maximum}}\left\{\hat{\alpha}^j y + \hat{\beta}^j\right\}}$

How do we accomplish this?

$$\boxed{\underset{y\in Y}{\text{Minimize}}\left[\underset{1\le j\le k}{\text{maximum}}\left\{\hat{\alpha}^j y + \hat{\beta}^j\right\}\right]}$$

By introducing a new (continuous) variable z, we can write the master problem as an "almost-pure" integer LP.

$$\begin{array}{ll} & \text{Minimize } z \\ \text{subject to} & \left\{\begin{array}{l} z \ge \hat{\alpha}^1 y + \hat{\beta}^1 \\ z \ge \hat{\alpha}^2 y + \hat{\beta}^2 \\ \quad\vdots \\ z \ge \hat{\alpha}^k y + \hat{\beta}^k \end{array}\right. \\ & y \in Y,\ z \text{ unrestricted} \end{array}$$

Solving the Subproblems

The *Dual Simplex* Method should be used in solving the subproblems...

The optimal dual solution $\hat{u}$ of the previous subproblem will still be feasible in the next subproblem, and can be used as the initial basic feasible solution of the dual, whereas using the *primal* simplex method would generally require a Phase-One procedure with artificial variables in order to obtain an initial basic feasible solution.

Solving the Subproblems

Use of the Dual Simplex Method yields another "bonus":

Each dual-feasible solution encountered during the solution of a subproblem can be used to generate another linear support, thereby improving the approximation of the function v(y)

That is, multiple supports can be added at each iteration of Benders' algorithm!

Consider again our example:

$$\begin{array}{ll} \text{Minimize} & 18x_1 + 8x_2 + 20x_3 + 8y \\ \text{subject to} & 3x_1 + x_2 + x_3 + 2y \geq 6 \\ & x_1 + x_2 + 4x_3 + y \geq 10 \\ & x_j \geq 0,\ j=1,2,3,4 \\ & y \in \{0, 1, 2, 3, \ldots 12\} \end{array}$$

Iteration #1 Let $\bar{y} = 0$ be our initial "guess"

We must next evaluate v(0) by solving the LP with y=0.

Subproblem Evaluate v(0)

$$v(0) \quad \max 6u_1 + 10u_2$$
$$\text{s.t.} \begin{cases} 3u_1 + u_2 \leq 18 \\ u_1 + u_2 \leq 8 \\ u_1 + 4u_2 \leq 20 \\ u_1 \geq 0,\ u_2 \geq 0 \end{cases}$$

The maximum occurs at the extreme point (4,4), which we will label $\hat{u}^1$

Our initial approximation for the function v is

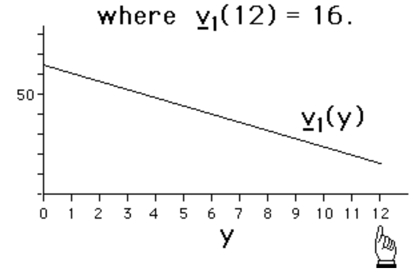
$$\underline{v}_1(y) = 8y + \max\ (6-2y)\hat{u}_1^1 + (10-y)\hat{u}_2^1$$
$$= -4y + 64$$

Note that $v(0) = \underline{v}_1(0) = 64$

That is, the approximation is exact, $v(y) = \underline{v}_1(y)$ for $y=0$.

Solving partial Master Problem minimize $\underline{v}_1(y) = -4y + 64$ s.t. $y \in \{0,1,2,3,\ldots 12\}$

The minimum occurs at $\bar{y} = 12$ where $\underline{v}_1(12) = 16$.



Subproblem Evaluate v(12), i.e.

$$v(12) = 8\times 12 + \max\ (6-2\times 12)u_1 + (10-12)u_2$$

$-18u_1 - 2u_2$

$$\text{s.t.} \begin{cases} 3u_1 + u_2 \leq 18 \\ u_1 + u_2 \leq 8 \\ u_1 + 4u_2 \leq 20 \\ u_1 \geq 0,\ u_2 \geq 0 \end{cases}$$

The maximum occurs at the extreme point (0,0), which we will label $\hat{u}^2$

Stopping Criterion $v(12) = 96 > 16 = \underline{v}_1(12)$, so we do <u>not</u> terminate.

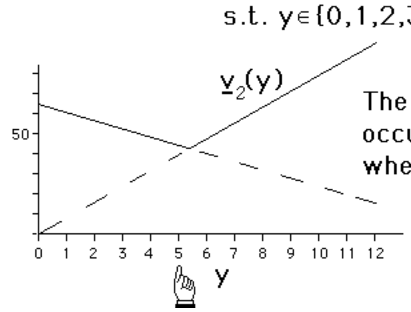
Adding a linear support to our approximating function $\hat{u}^2 = (0,0)$

$$\hat{\alpha}^2 y + \hat{\beta}^2 = 8y + (6-2y)\hat{u}_1^2 + (10-y)\hat{u}_2^2$$
$$= 8y$$

and so we obtain the new approximation

$$\underline{v}_2(y) = \max\ \{-4y+64,\ 8y\}$$

Solving partial Master Problem Minimize $\underline{v}_2(y) = \max\ \{-4y+64,\ 8y\}$ s.t. $y \in \{0,1,2,3,\ldots 12\}$



The minimum occurs at y=5, where $\underline{v}_2(5)=44$

Subproblem Evaluate v(5), i.e.

$$v(5) = 8\times 5 + \max\ (6-2\times 5)u_1 + (10-5)u_2 = -4u_1 + 5u_2$$
$$\text{s.t. } 3u_1 + u_2 \leq 18$$
$$u_1 + u_2 \leq 8$$
$$u_1 + 4u_2 \leq 20$$
$$u_1 \geq 0,\ u_2 \geq 0$$

The minimum value is 65, achieved at the extreme point (0,5), which we label $\hat{u}^3$

Stopping Criterion $\underline{v}_2(5) = 44 < 65 = v(5)$ so we cannot terminate

Adding a linear support to our approximating function

$\hat{u}^3 = (0,5)$

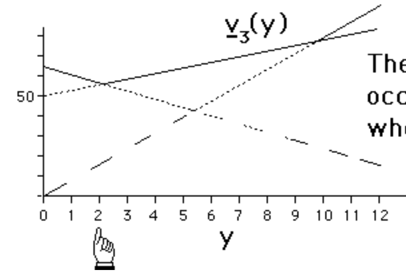
$$\hat{\alpha}^3 y + \hat{\beta}^3 = 8y + (6-2y)\hat{u}_1^3 + (10-y)\hat{u}_2^3$$
$$= 3y + 50$$

and so we obtain the new approximation

$$\underline{v}_3(y) = \max\{-4y+64,\ 8y,\ 3y+50\}$$

Solving partial Master Problem

Minimize
$\underline{v}_3(y) = \max\{-4y+64,\ 8y,\ 3y+50\}$
s.t. $y \in \{0,1,2,3,\ldots 12\}$



The minimum occurs at y=2, where $\underline{v}_3(2)=56$

Subproblem Evaluate v(2), i.e.

$$v(2) = 8\times 2 + \max\ (6-2\times 2)u_1 + (10-2)u_2 = 2u_1 + 8u_2$$
$$\text{s.t. } 3u_1 + u_2 \le 18$$
$$u_1 + u_2 \le 8$$
$$u_1 + 4u_2 \le 20$$
$$u_1 \ge 0,\ u_2 \ge 0$$

The minimum value is 56, achieved at both the extreme points $\hat{u}^1=(4,4)$ and $\hat{u}^3=(0,5)$

Stopping Criterion $\underline{v}_3(2) = 56 = v(2)$ so we can now terminate!

Suboptimizing the Partial Master Problem

Benders' master problem was to choose $y \in Y$ so as to

$$\text{Minimize } \underline{v}_k(y)$$

where $\underline{v}_k(y)$ is the current approximation to v(y), i.e.,

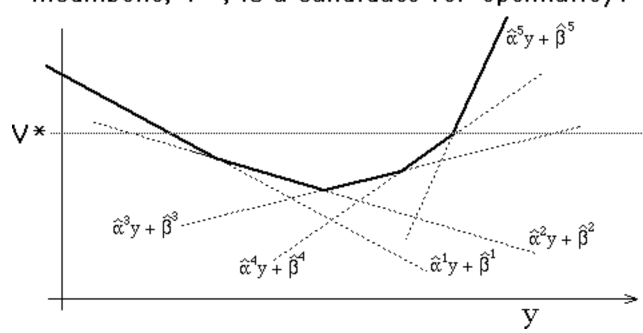
$$\underline{v}_k(y) = \underset{1\le j\le k}{\text{maximum}}\left\{\hat{\alpha}^j y + \hat{\beta}^j\right\}$$

This is accomplished by solving to optimality the (almost-pure) integer LP:

$$\begin{array}{ll}\text{Minimize } z & \\ \text{subject to} & \left\{\begin{array}{l} z \ge \hat{\alpha}^1 y + \hat{\beta}^1 \\ z \ge \hat{\alpha}^2 y + \hat{\beta}^2 \\ \vdots \\ z \ge \hat{\alpha}^k y + \hat{\beta}^k \end{array}\right. \\ & y \in Y,\ z \text{ unrestricted}\end{array}$$

by an implicit enumeration (branch-&-bound) algorithm. This is generally the most costly part of the total computation!

Any y such that $\underline{v}_k(y)$ is less than the incumbent, V*, is a candidate for optimality.



Rather than optimizing the master problem, therefore, we might seek only a feasible solution to the "pure" integer LP:

$$\left\{\begin{array}{l} \hat{\alpha}^1 y + \hat{\beta}^1 \le V^* \\ \hat{\alpha}^2 y + \hat{\beta}^2 \le V^* \\ \vdots \\ \hat{\alpha}^k y + \hat{\beta}^k \le V^* \end{array}\right.$$
$$y \in Y$$

This modification to Benders' algorithm will result in significant savings in CPU time.

Embedding Benders' Algorithm in an Implicit Enumeration

This is a modification of Benders' algorithm with suboptimization of the Master Problem

Suboptimizing the master problem has been accomplished when reaching a terminal node of the enumeration tree.

Find $y \in Y$ satisfying

$$\left\{\begin{array}{l} \hat{\alpha}^1 y + \hat{\beta}^1 \le V^* \\ \hat{\alpha}^2 y + \hat{\beta}^2 \le V^* \\ \vdots \\ \hat{\alpha}^k y + \hat{\beta}^k \le V^* \end{array}\right.$$

The next partial master problem differs from the previous one in that

- it has an added constraint
- the right-hand-side V* might be lower (if the incumbent has been replaced by the solution of the subproblem just solved)

Each of these changes to the system of inequalities reduces the feasible region of the system....

Hence, any portion of the enumeration tree which was fathomed during the previous tree search remains fathomed when the subsequent tree search begins.

That is, the enumeration can be "restarted" at the terminal node which had been reached in the previous Master Problem solution.

The enumeration tree is completely searched only once during the entire algorithm!