An Updated Rosen Partitioning Algorithm For Nonlinear Programming

by

Richard Edahl

EDRC 05-47-90 *

AN UPDATED ROSEN PARTITIONING ALGORITHM

FOR NONLINEAR PROGRAMMING

Richard Edahl
Engineering Design Research Center
Carnegie-Mellon University
24 January 1990

Abstract[1]

Rosen's Partitioning Algorithm for nonlinear programming was developed using a gradient projection method for the master problem. Since then, developments in NLP algorithms have shown the power of the Han-Powell successive quadratic programming algorithm. Here, the formulas necessary for using the Han-Powell algorithm for the master problem of Rosen's algorithm are derived. Sensitivity results are used to handle the 'crossover' problem.

Keywords: nonlinear programs, convex programs, large-scale systems, nonlinear algorithms, linear programs

[1]Correspondence should be sent to EDRC,Carnegie-Mellon University, Pittsburgh PA, 15213.

1. Introduction

This **paper examines** an update of Rosen's algorithm (Rosen [5], Lasdon [3]). Rosen's is a partitioning or **decomposition algorithm tor large-scale systems,** utilizing a single master problem and one or possibly several **subproblems.** In **the** algorithm's original form (described in Section 2), the subproblems were to be linear **programs and the master problem was to** be solved **by a** gradient projection method. Probably one of the most popular nonlinear programming algorithms today is the Han-Powell successive quadratic programming algorithm (Han [2], Powell [4]). Section 3 gives the formulas and results necessary to solve the master problem using the Han-Powell algorithm instead of a gradient projection algorithm. One of the messiest parts of Rosen's original algorithm was how to deal with cross-over problem. A significantly simpler approach is suggested here in Section 4.

2. Rosen's Algorithm

Rosen's partitioning algorithm was originally intended for problems of the form (see [3,5]):

$$P: \min_{xy} d(y) + cx \quad Ax + b(y) \le 0 \quad g(y) \le 0 \tag{1}$$

where the problem P is linear in the x variables and non-linear in the y variables. The basic idea of the algorithm is to solve a nonlinear master problem (in y) with linear subproblems (in x). The linear subproblem associated with P is given by:

$$LP(y): \min_x cx \quad Ax + b(y) \le 0 \tag{2}$$

It may be the case that the linear program LP(y) can be broken into several independent (with respect to the x variables) and smaller linear programs. For simplicity of notation, this dividing step will not be done.

Let the solution to LP(y) be x*(y), and let c*(y)-cx*(y). Define the master problem:

$$P^*: \min_y d(y) + c^*(y) \quad g(y) \le 0 \tag{3}$$

The problems *P* and *P** are clearly equivalent. c*(y) can be expected to be a piecewise smooth function with each optimal basis of LP(y) determining the region or patch of smoothness. Rosen's Algorithm optimizes *P** over a given patch, and then decides whether local (or global if the entire problem is convex) optimality has been achieved or to move to an adjacent patch. Use is made of perturbation results in linear programming to aid in this process. The algorithm is shown to be globally convergent for the case where A and c **are** independent of y; b,d and g are convex in y; and certain differentiability and regularity conditions are **met. The** modifications suggested here do not affect the range of problems for which Rosen's Algorithm is convergent. In all that follows, the above assumptions on y,b,d,g,A, and c are made. Additionally, a non-degeneracy assumption for P is made.

Suppose that for a point y°, LP(y°) has been solved with a resultant x° solution and a set of tight basic constraints and a set of non-basic constraints:

$$Bx^0 + b_B(y^0) = 0$$

$$Nx^0 + b_N(y^0) \le 0 \tag{4}$$

Define $x_{fl}(y)$ and X, by:

$$x_B(y) = -B^{-1}b_B(y)$$

$$X_B = -cB\sim^{x} \tag{5}$$

Xj(y) is the optimal solution x*(y) to LP(y) as long as it is feasible, that is, as long as it satisfies the inequality constraints in Equation 4. Now substitute Equation 5 for x*(y) ink) Equation 3 for the restricted region to get:

$$P'(B): \min_{y} d(y) + \bar{\lambda}_B b_B(y)$$

$$-NB^{-1}b_B(y) + b_N(y) \le 0$$

$$8(y) \wedge 0 \tag{6}$$

That is, y is minimized over the region for which B is a feastole basis for the linear subproblem. The algorithm is to carry out this restricted optimization, and then see whether the solution is optimal to the full problem, or whether an adjacent patch should be entered.

Define

$$\lambda_B = \bar{\lambda}_B - \lambda_N N B^{-1} \tag{7}$$

$$\lambda = (\lambda_B, \lambda_N)$$

The Lagrangian for P'(B) is:

$$\begin{aligned} L_B(y,\lambda_N,\mu) &= d(y) + X_B b_B(y) + \lambda_N(-NB^{-1}b_B(y) + b_N(y)) + \mu g(y) \\ &= d(y) + Xgbgiy) + X\wedge f/y) + ng(y) \\ &= L(y,\lambda,\mu) \end{aligned} \tag{8}$$

The optimal solution y_B* to f (B) is optimal to the original problem iff $X_{fl} s 0$. If one of the components of X_B is negative, then the corresponding constraint can be pivoted into the basis with an expected improvement. It had been previously suggested that an auxiliary LP be solved to determine these pivots, but a possible improvement is given below in Section 4.

3. Han-Powell Formulation

When first introduced, Rosen's algorithm utilized a Gradient Projection algorithm for solving the restricted optimization problem P'(B) of equation 6. More recently, successive quadratic programming algorithms such as the Han-Powell algorithm have been advocated for solving nonlinear programming problems. Here we consider such an implementation.

As also for the Gradient Projection algorithm, the Han-Powell algorithm requires the computation of the function values and gradients for the objective function and constraints. Consider P'(B). The gradients of the objective function and constraints are given by:

$$\nabla_y d(y) + \bar{\lambda}_B \nabla_y b_B(y)$$

$$-NB^{-1}\nabla_y b_B(y) + \nabla_y b_N(y)$$

$$\nabla_y g(y) \tag{9}$$

These are used in a sequential quadratic programming algorithm of the following type:

$$\min_{\Delta y} \ \{\nabla_y d(y) + \lambda_B \nabla_y b_B(y)\}\Delta y + 1/2\, \Delta y^t M \Delta y$$

$$-NB^{-1}b_B(y) + b_N(y) + \{-NB^{-1}\nabla_y b_B(y) + \nabla_y b_N(y)\}\Delta y \le 0$$

$$g(y) + \nabla_y g(y)\Delta y \le 0 \tag{10}$$

Where M is an approximation to

$$\begin{aligned}\nabla_{yy} L_B(y) &= \nabla_{yy}\{d(y) + [\bar{\lambda}_B - \lambda_N NB^{-1}]b_B(y) + \lambda_N b_N(y) + \mu \nabla_{yy} g(y)\} \\ &= \nabla_{yy} L(y,\lambda,\mu)\end{aligned} \tag{11}$$

Since this Hessian is independent of the basis B, it is clear that should the basis change (e.g. after crossing over into another patch), the old M is usable as a starting estimate. That is, suppose that y^* is on the border between two or more patches, implying that it is the solution to the corresponding P'(B) restricted problems. The same Hessian estimate, M, is then appropriate for any of these different restricted problems. Hence, jumping from one patch to another (and possibly back) should not require restarting the algorithm (i.e. setting M to I).

M is generally constructed by using a variable metric rank-two update formula. For this, one needs to know y^0 and y^1 (the old and new y estimates respectively), λ^1 and μ^1 (the new dual variable estimates), $\nabla_y L(y^0,\lambda^1,\mu^1)$, and $\nabla_y L(y^1,\lambda^1,\mu^1)$. $(y^1,\lambda_N{}^1,\mu^1)$ is given by the solution to the quadratic programming problem, and $\lambda_B{}^1$ can be found from Equation 7.

4. Changing Bases

Consider a slightly different but equivalent version of P:

$$P'': \min_{x,y,s} d(y) + cx$$

$$\begin{aligned} &Bx + b_B(y) + s = 0 \\ &Nx + b_N(y) \le 0 \\ &g(y) \le 0 \\ &s \ge 0 \end{aligned} \tag{12}$$

Define $x_B(y,s)$ as the solution to:

$$Bx + b_B(y) + s = 0$$

$$(x_B(y,s) = -B^{-1}[b_B(y) + s]) \tag{13}$$

P'' is then equivalent to

$$P''(B): \min_{y,s} d(y) + \bar{\lambda}_B b_B(y) + \bar{\lambda}_B s$$

$$\begin{aligned} &-NB^{-1}b_B(y) - NB^{-1}s + b_N(y) \le 0 \\ &g(y) \le 0 \\ &s \ge 0 \end{aligned} \tag{14}$$

Define the parametric program

P"(fij): min dfy) + X#$_B$(y) + $\bar{\lambda}_B s$

7

$-NB^{-1}b_B(y) - NB^{-1}s + b_N(y) \le 0$

$8(y) \le 0$

for 5*0 **(15)**

Solving />"(B,0) (which is identical to *P*(B))* by the Han-Powell method is a possible improvement over Rosen's original suggestion of using a gradient projection method. However, the above method enables a further simplification. It is a relatively simple matter to compute the gradient of optimal value function of /"'(B.O) with respect to s. If any of the components of this gradient are negative, then an increase in the corresponding s_i will cause a decrease in the value of the program. This change in s can be used to drive a change in the y variables, automatically causing a change of basis in the linear programming problem, LP(y)(see [1]). (Rosen's method, as given in Lasdon [3] is to change the basis artificially. That is, to force the corresponding constraint out of the basis by pivoting. The method described here seems more natural.)

Suppose />"(B,0) has been solved yielding an optimal y(0). By the non-degeneracy assumption, the gradient of the optimal value function with respect to s is simply (see Fiacco [1])

$$\bar{\lambda}_B - \lambda_N(0)NB^{-1} = \lambda_B(0) \quad (16)$$

and the gradient of y(s), with respect to s, at 0 is given by the solution to

r 0 i rV$_{yy}$Z° Vh_y<> V$_y$g°] r V(0> i

i i i i i i

\NB-^{1} I =1 V$_V$A° | |VA<(0) I

I I I I I I

L o J Ly^° J Lvy(0) J **(17)**

$\nabla_y h = \nabla_y\{-NB^{-1}b_B(y(0))+b_N(y(0))\}$

where

$\nabla_{yy}L^0$ = *Vyy{d(y(0))+\b$_B$(y(0))* $+ \lambda_N b_N(y(0)) + \mu g(y(0))\}$

h° includes only the tight constraints from P"(B,0).

A simple procedure for changing the basis in *P* is to perform an elementary steepest descent step in s on P"(0), keeping at 0 all components of s with a non-positive value for \$_B$. Define \+$_B$ as the vector with the same values of X_B for positive components and 0's for the negative components. The following is a descent direction for the master problem:

Ay = V ^ O ^ x y (18)

A crude line search over positive S could be used, but this is not necessary since the point of the perturbation is to change the basis so further improvement is possible.

4.1. Example

The above modifications to Rosen's method were applied to the following problem.

i a

min-Uj - 2xj+3(yr5)2 + (yj-2)^2 + 2C3-8)^2

$$x_1 + x_2 + y_1 + y_2 + y_3 \le 20$$
$$x_1 + 2x_2 + y_2 + .1y_3 \le 19$$
$$-x_1 + x_2 + 2y_1 + y_3 \le 18$$

$-x_x$ £ 0 -*2 * 0

The sub-problem is given by:

LP(y) m*>i-Ix$_I$-2x$_2$

rowl	$x_1 + x_2 + \{ y_1 + y_2 + y_3 - 20\} \le 0$
row2	$x_1 + 2*2 + \{y_2 + .1y_3 - 19\} \le 0$
row3	$-x_1 + x_2 + \{2y_1 + y_3 - 18\} \le 0$
row4	$-x_1 \le 0$
row5	

The initial y and the resultant x from LP(y) are

Basis is ***(row2, rowj)***
y, = 2. x, = 1.1
y_2 = 1. x_2 = 8.1
y_3 = 7.

$\lambda_B = (1, 0)$

There is an alternate optimal solution of x$_1$-2.7, x$_2$«7.3 with basis of *(row\\,* ***row*3).**

B = r 1 2 1 L-I 1 J *B*$^{-1}$ = r 1/3 -2/3 1 L1/3 1/3J

r i ii
iV = 1-1 01
L o -l j

ro i .in>,i r 191
$b_B(y)$= I 11.% I - I I
L2 0 1 JIy$_3$J L18 J

n ı 1]Γy,] Γ20]
$b_N(y)$ = I o o o 11% I - I o I
Lo o o JLy$_3$ J L o J

The **general master** problem is

P\B)min *d(y)* + $\lambda_B b_B(y)$
y

-*NBr^j* $b_B(y) + b_N(y) \le 0$

For the current basis, this becomes:

P'(B)mi* 1 5&J-5)^2 + $(y_2$-2f 3 + Hyf*? + y_2--ly_3- 19
y

r 1.667 0.333 1.267 "ll" yj l"-13.33 1
1-1.333 0.333 -0.633 ll y_2 l + l 5.67 l 10
L 0.667 0.333 0.367 JL y_3J L-12.33 J

The solution to the master problem gives the new y and X^ vectors and updated x (by Equation 5 and X_B (by Equation 7) vectors:

y, = 2.2333185 x, = 1.50499
y_2 = 1.2233471 Xj = 7.77254
y_3 = 7.2658054

X = (1.6600, -.1067,0.5533,0.,0.)

Since X^ is negative, it should be pivoted out of the LP basis. The direction for the Ay vector is given by the solution to:

r o i [Q v<h° ir ∇,y(0)]
| | = | || |
LNB⁻¹J L ∇,g⁰ JL∇,λ_N'(0) J

$\Delta y = \nabla_x y(0)_B \lambda^{*}{}_B{}^{t}$

To 1 \Q ,V'A° i∇ (0)λ v]
l 1 = 1 ll l
UVB-'XV J Lv/ JLv/V(°) -l

Plugging the values for the current point and basis,

NB~^{x}X*$_B{}^t$ = \ 2/3 -1/3 1 f-.106661
l -1/3 2/3 l l l
L-l/3 -1/3 J L 0 J

TO 1 T 1 0 0 5/3]⌈∇ ^(0)]
10 1 = 1 0 2 0 1/3 11 ∇>$_2$(0) l
l 0 1 1 0 0 3 3.8/3 ll ∇,y_3(0) l
L.07111J L5/3 1/3 3.8/3 0 JL ∇,λ_1(0) J

Ay* = (.03513, .00358, .00897)

Only the first row of N was selected since the other two corresponded to loose constraints in P'(B).

A value of 8-.1 **was** used, resulting in a new y and x values of:

Basis is *(rowl,row3)*
y, = 2.23335 JC, = 1.50500
y_2 = 122335 *2 = 7.77248
y_3 = 726581

X$_{/>}$ = (1.5,.5)

Since the constraints are all linear, any small value of a would have the same result.

$$B = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \qquad B^{-1} = \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & 1/2 \end{bmatrix}$$

$$N = \begin{bmatrix} 1 & 2 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$b_B(y) = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} - \begin{bmatrix} 20 \\ 18 \end{bmatrix}$$

$$b_N(y) = \begin{bmatrix} 0 & 1 & .1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} - \begin{bmatrix} 19 \\ 0 \\ 0 \end{bmatrix}$$

The new master problem is

$$P'(B)\ \min_{y}\ \tfrac{1}{2}(y_1-5)^2 + (y_2-2)^2 + \tfrac{3}{2}(y_3-8)^2 + \tfrac{5}{2}y_1 + \tfrac{3}{2}y_2 + 2y_3$$

$$\begin{bmatrix} -0.5 & 0.5 & 0.0 \\ -2.5 & -0.5 & -1.9 \\ 1.5 & 0.5 & 1.0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} + \begin{bmatrix} -1 \\ 20 \\ -19 \end{bmatrix} \le 0$$

New y and λ_N vectors from P'(B) and resultant x and λ_B vectors:

$y_1 = 2.5000022 \quad x_1 = 1.62499$
$y_2 = 1.2500134 \quad x_2 = 7.29165$
$y_3 = 7.3333409$

$\lambda = (1.5, 0., 1.5, 0., 0.)$

Since all of the dual variables are non-negative, this last solution is optimal. The above was computed using an inexact algorithm for solving the quadratic programming problem. The exact solution is:

$y_1 = 5/2 \quad x_1 = 13/8$
$y_2 = 5/4 \quad x_2 = 175/24$
$y_3 = 22/3$

$\lambda = (1.5, 0., 1.5, 0., 0.)$

REFERENCES

[1] Fiacco, Anthony V.
Introduction to Sensitivity and Stability Analsis in Nonlinear Programming.
Academic Press, New York, New York, 1983.

[2] Han, S-P.
Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems.
Mathematical Programming 11:263-282, 1976.

[3] Lasdon, L.S.
Optimization Theory for Large Systems.
The Macmillan Company, New York, 1970.

[4] PoweH, MJ.O.
A Fast Algorithm for Nonunearfy Constrained Optimization Calculations.
Technical Report, University of Cambridge, June, 1977.

[5] Rosen, J.B., and Ornea.J.C.
Solution of Nonlinear Programming Problems by Partitioning.
Management Science 1O(1):16O-173,1963.