On the Convergence Rate of A Branch-and-Bound Algorithm for Unconstrained Problems

by

Richard Edahl ^

'A

EDRC 05-45-89^

ON THE CONVERGENCE RATE OF A BRANCH-AND-BOUND ALGORITHM FOR UNCONSTRAINED PROBLEMS

Richard Edahl
Engineering Design Research Center
Carnegie Mellon
30 October 1989

This work has been supported by the Engineering Design Research Center, an NSF Engineering Research Center.

Abstract

Branch-and-bound methods have been used with mixed results for globaF optimization problems. The asymptotic convergence rate of a theoretical branch-and-bound algorithm for twice-differentiable unconstrained optimization is examined to shed light on when branch-and-bound methods can be expected to be successful. The Generalized Weber Problem from location theory is considered as a special case, with two different lower-bounding functions examined.

1. Introduction

The global optimization problem is given by

minflx)
x∈S

where the continuous objective function/.R^n-»R, and the feasible region is the compact set *Se* R* usually defined by a system of inequalities and equalities:

h£x) £ 0 Jfc=l,2,..^Tl

Since for this problem local methods frequently fail due to non-globally optimal local minima, several other types of algorithms have been suggested. Among these are probabilistic methods (see e.g. efforts reported in Dixon[3]), grid methods (see e.g. Shubert[12], Meewella and Mayne [10], and Dixon [3]), and branch-and-bound methods (see e.g. Mitten [11] and Horst [7]). We consider here problems that are essentially unconstrained with S a simple region such as an n-dimensional rectangle.

A branch-and-bound algorithm, as described by Horst, partitions the feasible set S into a collection of regions (e.g. hyper-rectangles or simplices), and then computes a lower bound of the objective function for each region (and within the constraints if any). Then one of the regions (generally the one with the lowest lower bound) is further partitioned into smaller regions, with new lower bounds computed for these new regions. This procedure is followed until a satisfactory convergence has been achieved. Horst gave some sufficient conditions for convergence, and gave versions of the branch-and-bound algorithm that have been or could be implemented for concave programs (convex region S, concave objective function f) and variants (e.g. some constraints $h(x)_k$£0 not convex).

Related is the grid algorithm as given by Shubert (for the 1-dimensional case) and Meeweite and Mayne (for higher dimensions). They considered the problem for which S is an n-dimensional rectangle, and for which a Lipschitz number L is known for f over S. The function f is evaluated at the points on a grid, and these values, combined with the Lipschitz number, s used to a compute a lower bound for f over the individual cells or n-dimensional rectangles (each of which has grid points for vertices). The best cell is partitioned, but the new small ceils are generally not all of the same size. The reported results with these algorithms has not been encouraging.

Branch-and-bound algorithms have been used for certain location problems. One example is the BSSS (Big Square-Small Square) algorithm given in Hansen et. al. [6]. It was constructed for use in location problems with polyhedral constraints (including 'or' constraints), and used a lower-bounding function that is related to one using the Lipschitz number (see Section 4). A somewhat more complex lower-bounding function was used in Edahl [4] for a slightly more restricted class of unconstrained location problems. It demonstrated significantly better convergence than did the Hansen algorithm for the same unconstrained location problems.

While initially appealing, branch-and-bound and grid algorithms seem to be generally affected by a very bad convergence rate, particularly in higher dimensions. Here we examine this convergence rate for unconstrained problems and suggest sufficient conditions on the bounding function for a more satisfactory convergence rate. A theoretical branch-and-bound algorithm is given in Section 2. In Section 3, the convergence rate of this algorithm is analyzed for strictly convex quadratic functions. The quadratic case would give the asymptotic rate of convergence for a general locally strictly convex and twice differentiate function. In Section 4, the convergence rates of two different lower-bounding functions for location problems (Hansen's algorithm and Edahl's algorithm) are analyzed.

2. Branch-and-Bound Algorithm

A theoretical branch-and-bound algorithm is given in this section. Where both the branching (including partitioning) and the bounding functions are normal, the selection rule for the region to be subdivided is chosen for its theoretical rather than practical value. For purposes of analysis here, the set S is primarily for convenience, for it is the asymptotic behavior of algorithm in the unconstrained case that is of interest here. It is expected that the optimal solution lies in the interior of S, which allows us to assume that S is some convenient region containing an open neighborhood of the optimal solution. While the normal selection rule is to chose the cell with the lowest lower-bounding function value, here the rule is to select and divide all active cells simultaneously.

2.1. Partitioning and Branching

Two common choices for regions to be bounded have been n-dimensional rectangles and simplices (see e.g. [7]). Here, the rectangle or cell oriented along the axes is selected as the basic region. When a cell is sub-divided, it results in 2^n new cells all of the same size. This differs from the dividing method given in Property 4.4 of Horst [7] in that there, a single cell is divided into only two equal sized rectangles by selecting the midpointsLOf..p_fJhe longest^edges (assuming all dimensions of the rectangle are different) as the common vertices of the two new cells. The sub-dividing here is much quicker (and would be the result of n of Horst's subdivisions). Below this is formalized.

Definition 1: R is a <u>cell</u> in R" if

3ye R" and ct€ R",a>0 such that
R = R(y,a) m { xs R^n **: |y-*| ≤ *of,* /=1....*n)***

A <u>partitioning</u> P of a set S^R" is a collection of nonoverlapping cells *[Rp}p€* /> whose union covers S, where *P* is an index set. That is, *{Rp)pe P* satisfies the following three conditions:

(a) $R_p \cap R_q = \partial R_p \cap \partial R_q \quad \forall p,q \in P, p \neq q;$

(b) $S \cap R_q \neq 0, \ \forall p \in P$; (c) $S \subseteq \cup_{p \in P} R_p$.

For the analysis of the branch-and-bound algorithm, it will be assumed that all cells in a given partitioning are of the same dimensions.

Definition 2: Let a be a fixed strictly positive n-vector and 8 a positive scalar. Define:

the cell radius

$$r(R(y,\delta\alpha)) = \{\sum_i^n (\delta\alpha^i)^2\}^{\frac{1}{2}} \qquad B(a) = *R(y,a))$$

the cell volume

$$V\{R(y_f fa)\} = (25)^n na'*; \qquad C(a) = V\{R(y_t a)\}$$

Definition 3: P is a 8a-cell partitioning of S if

1) It satisfies the requirements of a partitioning from Definition 1

2) For each $p \in P$, $\exists\, y_p \in R^n$ such that $R_p = R(y_p, S a)$

We need the idea of a branching or dividing function that takes a single cell and generates 2^n new cells. Consider a general cell R. Let

c(m) = (cHm)9...SKm))

be the binary representation of m-1, so that

$$m-1 = \sum_{i=0}^{n-1} 2^i c^{n-i}(m)$$

Definition 4: Let R be an arbitrary cell in R^n. Then the cell dividing function •={<h.$2-- ^2"} generates 2^n cells $\Phi_m(R)$, m=1,2....2^n defined by:

$$\phi_m(R) = \{x \in R^n : |x^i - y^{m,i}| \leq \alpha^{m,i}, \ i=1,\ldots,n\}$$

where

$$\alpha^{m,i} = \alpha^i/2; \quad y^{m,i} = y^i + \alpha^i(c^i(m)-1/2)$$

Note that

i) $\cup_{m=1,\ldots,2^n} \phi_m(R) = R$

ii) $\phi_l(R) \cap \phi_m(R) = \partial\phi_l(R) \cap \partial\phi_m(R)$ for $l \neq m$.

2.2. Lower-Bounding Functions

Branch-and-bound algorithms operate by dividing the region into cells and finding lower bounds for f(x) in each of the cells. It is necessary that for each subcell R_p, the lower bound $F(R_p)$ satisfy

$$F(R_p) \leq \min_{x \in R_p} f(x)$$

The lower bounds on the cells in a partition can be used to give an overall lower bound estimate for f over S (by taking the minimum of F(Rp)), to eliminate cells from further consideration (if $F(R_p)$>f* where f* is the value of the best feasible point found so far), and to direct the further division of the cells (e.g. divide the cells with the lowest $F(R_p)$'s).

For a given partition, lower-bounding function, and best current objective function value r it can be shown that cells for which $F(R_p)$£f* cannot contain a better solution. Such cells can be discarded from further consideration. What is left will be a set of 'active cells' and an 'active region'. Formally,

Definition 5: For a given cell partition P, a lower-bounding function F, and a given best (lowest) upper bound f^*_t the active cell index set I(F,f*) is given by

I(Ff)=IpeP: F(I?p<I}

and the active region by

1^={x: xeR_p for some *pel(Ff))*

The cardinality of *I(ff)* is designated by *\I(Ff) \.*

In order for this sort of algorithm to work, some restrictions on F(R) are necessary. A Consistency Condition for the lower-bounding function is given by Horst [7] as a sufficient condition for the branch-and-bound algorithm to converge. In terms of 8cc-cells, this Consistency Condition for unconstrained optimization reduces to:

Let *[y_p] -*y* and {5_p} -»0+ be arbitrary sequences. Then F satisfies the Consistency Condition if {F(R(y_p,5_p<x)} *->fly)*

Three classes of lower-bounding functions are given below. Each trivially satisfies the Consistency Condition if f is differentiate over S, and hence the branch-and-bound algorithm will converge for such bounding functions.

The simplest lower-bounding function utilizes the Lipschitz number for the function f(x) over the region S. A Lipschitz number as used here satisfies:

\f(y)-Ax)\≤L\\y-x\\$_2$ VxjeS

Definition 6: Let f be a function defined on a set S with Lipschitz number L Then F(R) is a Lipschitz Lower-Bounding Function (llbf) for f over S if

$$F(R) = f(y(R)) - Lr(R)$$

where y(R) is the center of the cell R, and r(R) is the cell radius.

One of the earliest grid algorithms was by Shubert [12], and used the Lipschitz number in a more complex manner to construct a lower-bounding function. He did this for the 1-dimensional case, and Meewella and Mayne[10]) extended this to higher dimensions. Both algorithms operated by dividing the feasible region S into cells and evaluating the function f(x) on the vertices of all of the cells. Letting L being a Lipschitz number for f over S, a lower bound for f over a cell R could be found by solving

min max {fiXf^-LWx-x^}
x€RxI_k€R

Unlike the branching function defined in the previous section, the solution x, to this problem would be used to define 2^n new cells; the current cell containing x, is divided into new cells with X| being the vertex common to all of these new smaller cells. This branching step results in new cells usually of all different sizes.

One problem with using Lipschitz number is that is does not change for cells near the optimal solution. If f is C^1 over S, a valid L can be found by

$$\overline{L} = \max_{xcS} ||V^)||_2$$

This follows since

$$|/(yH(*)| \leq \underset{x*S}{Max} \backslash V^vXy-*)| \quad \text{(By Taylor's Theorem. See e.g. Goldberg [5])}$$

$$\leq \underset{xaS}{Max} \|\nabla_x f(v)\|_2 \|y-x\|_2 = \overline{L} \|y-x\|_2$$

But f being C^1 over S implies that at the optimal solution the gradient of f is the 0 vector. Hence a Lipschitz number that is valid for all of S would be arbitrarily loose near the optimal solution. It seems natural to use a lower-bounding function similar to a llbf but with the constant L changing with the cell.

Definition 7: Let f be a differentiate function defined on a set S. Then F(R) is a Gradient Lower-Bounding Function (glbf) for f over S if

$$F(R) \; £ \; \min_{X \in R} ftx)$$

and 3 $K>0$ such that

$$f[y(R)) - F(R) \; £ \; Kr(R)\overline{L}(R) \quad \forall \; R \subseteq S$$

where $I(R) = \max_{X \in R} ||V/[JC)||_2$

Another powerful lower-bounding function can be found by first finding a sub-function Jhat bounds f from below over a cell, and then minimizing this function over the cell. The idea of using a sub-function has been used for concave programming problems (concave objective function, convex constraints) for example by McCormick [9] and Horst [7].

Definition 8: Let f be a differentiable function defined on a set S. Then F(R) is a *rj_* Lower-Bounding Function (r^2lbf) for f over S if there is a sub-function of f over R such that

$$F(R) = \min_{xeR} f\{x \backslash R)$$

where

$$f(x|R) \leq f(x) \quad \forall x \in R$$

and 3 $Q > 0$ such that

$$Ax)-Qr(R)^2 < Ax \backslash R) \quad \text{Vie/?} \quad \forall R \subseteq S$$

Such an f(x|R) is an r^2-sub-function of f over R.

2.3. Algorithm

Above were described a cell, a dividing function, and some lower-bounding functions. What remains to complete a description of a branch-and-bound algorithm is a selection rule for choosing what cells are to be divided and an upper-bound function for a cell. Horst showed in [7] that the algorithm would converge for an upper-bound function consisting of evaluating the function at any point x_p in the cell R_p and for any selection rule as long as after a finite number of steps the cell with the lowest F value were selected for division.

To examine the behavior of the branch-and-bound algorithm, we use the following algorithm. Instead of branching on the most promising cell or node, all active nodes are branched on simultaneously. Since it is not expected that the algorithm would converge in a finite number of iterations, we use the idea of an e-solution. We consider x* € S to be an e-solution if

$$f(x^*) - \varepsilon \leq f(x) \quad \forall x \in S.$$

The problem shall be deemed solved if, for a given e > 0, an e-solution is found.

ALGORITHM: (Assume e>0 is given)

Step 1 For given 5 > 0, let P** be a 5a-cell partitioning of S). Set f* = ∞

Step 2 For each $R_p e$ P**, compute F(Rp) and f(Xp). Set

$$ft = \min_{p\in P^{--}} fiXp) \quad (xt \text{ is } x_p \text{ solution})$$

$$F^* = \min F(R_p)$$

if ft < f\ set f*=ft and x*=xt. For each cell, if $F(R_p) \geq$ r, delete R_p from $P^{\delta\alpha}$. Test whether f*-e £ F\ If no, go to Step 3. Otherwise, Stop (x* is then an e-solution).

Step 3 Form partition P! by applying Φ to each R_p in P^{6**}. Set P^{5**} to P!, and 5 to 5/2. Go to Step 2.

3. Convergence Rate

We examine the case where f(x) is quadratic with a positive definite matrix M. Without loss of generality, assume there is no linear term (hence 0 is the unique global minimizer). We do this to analyze the behavior of the branch-and-bound algorithm near an optima, for near an isolated optima, a C^2 function is nearly a positive definite quadratic function. The method of analysis is to form an arbitrary P** partition and then find bounds on the number of active cells using the various types of lower-bounding functions from the previous section.

For this section, f is given by

fix) = xfMx

where M is a fixed positive definite matrix.

Definition 9: The ellipse E?by is defined by:

£7 = {*€R^n: $x^tMx \leq f$}

and its volume is designated by V{ET}. The volume of the unit ellipse is designated as *A =V{E ^*

Definition 10:

the maximal M distance from 0 to the cell

g{R(y,Sa)) = max ^Mfy D(ct) = *g{R(0,a)}*

$y \in R(y,\delta\alpha)$

the maximal gradient of f in an ellipse

*Lit) = max 2^y*MMy* *L° =* L(l)

$y \in E^t$

Some elementary results that will be used later are:

V{R(y,Sa)) = 5"C(a); *KR(yte))* = 55(a);

V{EY} = γ^nA; *g{R(0,5a))* = 8D(a);

L(t) = tL°; *[xfMx)$^{1/2}$* S *{(x-yyM(x-y)}^{m}* +{/Afy}$^{1/2}$;

The last relation, the triangle inequality, is valid since {x^x}^{172} is a legal norm.

The analysis below is for a given symmetric positive definite matrix M and a given cell shape a. To simplify notation, the argument a is suppressed from the functions B (unit cell radius), C (unit cell volume), and D (maximal M distance to unit cell with the origin as center).

The following lemma gives bounds for the number of cells, in a given partitioning, whose centers fall inside the given EY ellipse. It is used as the basis for theorems on bounds of the number active cells for the three lower-bounding functions given in the previous section.

Lemma 11: Let P be an arbitrary 5a-cell partitioning of R^*_f and for a given y, let *Kfiyi-))^^{1})* be the set of active cells. The number of active cells *\I(f(y()),f)* is bounded by:

$$\frac{A}{C}\left(\frac{\gamma}{\delta} - D\right)^n \leq |I(f(y(\cdot)),\gamma^2)| \leq \frac{A}{C}\left(\frac{\gamma}{\delta} + D\right)^n$$

Proof: Let y^5 = y-8D; •/- = y+ 6D. Then £^ c 7 ^ c ^ by Lemma 15 in the Appendix. Therefore,

$$V\{E^{\gamma^S}\} \leq V\{T^P_{\gamma^2}\} \leq V\{E^{\gamma^L}\}$$

$$\rightarrow$$

$$A(y\text{-}5D)^n \quad Z\&QKflyi)),?)] \quad ZA(y+ 5D)^n$$

$$\rightarrow$$

CO $\leq |I(f(y(\cdot)),\gamma^2)| \leq \frac{A}{C}(\frac{\gamma}{\delta} + D)^n$

Q.E.D.

The following theorem establishes bounds on the number of active cells after an iteration using a Lipschitz based lower-bounding function. In the theorem, because f is not Lipschitz over the entire space, confining the feasible region to compact set S is necessary in order to find a Lipschitz number for f. S is chosen to be an ellipse for convenience. However, using such a feasible region complicates the situation in that some part of the active region may be infeasible. The restriction on 5 ensures that this is not the case.

Theorem 12: Let Y^M>0 and L>T^ML° be given. Let P be an arbitrary 5a-cell partitioning of R* with

$$8S \frac{\{\gamma^M\}^2}{BL+2\text{-}fD}$$

Let S^E^1. Let F(R) be a Lipschitz Lower-Bounding Function with Lipschitz number L Then for any iteration (and all subsequent iterations) of the algorithm for which 5 satisfies the above condition, the following bounds on the number of active cells after that iteration hold:

$$\frac{A}{C}(\sqrt{\frac{BL}{\delta}} - D)^n \leq |I(F,f^*)| \leq \frac{A}{C}(\sqrt{\frac{BL}{\delta}+D^2} + D)^n$$

Proof: Let f* be the value of the best point after an iteration for which the hypothesis of the theorem is satisfied. Then

$$0 \leq / \leq \{5D\}^2$$

because f is positive and the farthest the best point can be from 0 in M-distance is g{R(0,5a)}=5D. Now cell R_p is active iff

$$F(R_p) = f(y_p) - \delta BL \leq f^*$$

$$\rightarrow$$

$$y_p{}^tMy_p = f(y_p) \leq f^* + \delta BL$$

Letting y equal to the right hand side of this last inequality, the condition becomes $y_p e$ &. *y* is bounded by

$$bBL \leq f \quad \leq 5SL+\{5D\}^2$$

Thus the conditions of Lemma 16 in the Appendix hold, so ^ >_ 7^1- = *y* + 5D and hence *E*^ c *E*^% which means the outer bounding ellipse is entirely in the feasible region. By Lemma 11, the number of active cells is bounded by

Co C $\frac{\gamma}{\delta} + D)^n$

Since the bounds are monotonic in y, plugging in the smallest value into the lower bound and the highest value in the upper bound will yield new valid bounds:

$$\frac{A}{C}\left(\frac{\sqrt{\delta BL}}{\delta}-D\right)^n \le |I(F,f^*)| \le \frac{A}{C}\left(\frac{\sqrt{\delta BL+\{\delta D\}^2}}{\delta}+D\right)^n$$
$$\rightarrow$$
$$\frac{A}{C}\left(\sqrt{\frac{BL}{\delta}}-D\right)^n \le |I(F,f^*)| \le \frac{A}{C}\left(\sqrt{\frac{BL}{\delta}+D^2}+D\right)^n$$

Q.E.D.

Asymptotically, as 5 approaches 0, the number of active cells becomes

$$|I(F,f^*)| = \frac{A\sqrt{BL}}{C}\delta^{-n/2}$$

Letting 5=5°2*^{H} where H is the iteration,

$$|I(F,f^*)| = \frac{A\sqrt{BL}}{C}(\delta^0)^{-n/2}2^{nH/2}$$

The ratio of the number of cells from the (H+1)th iteration to the Hth iteration is $2^{n/2}$$_f$ which means that the number of cells (and thus the number of function evaluations) increases exponentially with the number of iterations. After an iteration, the best function value f* and lowest lower-bound for a cell F* are bounded by

$$0 \le f^* \le \{5D\}^2$$

$$-\delta BL \le F^* \le \{\delta D\}^2 - \delta BL$$

Hence, the difference between the two is bounded by

55L-{5D}^{2} $\lesssim$ / - F * $\lesssim$ 5£L+{5£>}^{2}

which implies that the difference is linear in 5 or roughly halved for every iteration of the algorithm.

The following theorem gives bounds on the number of active cells should a Gradient Lower-Bounding Function be used.

Theorem 13: Let P be an arbitrary Sot-cell partitioning of R" Let f(x) be defined as x*Mx, and let F(R) be a Gradient Lower-Bounding Function for f over S. Then for any iteration of the algorithm, the following upper bound on the number of active cells after that iteration holds:

$$|I(F,f^*)| \le \frac{A}{C}(KBL^0+2D)^n$$

Proof: By Lemma 17 in the Appendix, R is active only if *y* e E? where Y = *6(KBL°+D).* By Lemma 11, the upper bound of the number of active cells is

$$|I(F,f^*)| = |I(f(y(\cdot)),\gamma^2)| \le \frac{A}{C}(KBL^0+2D)^n$$

Q.E.D.

Finally, bounds are found for the number of active cells using an r^2 lower-bounding function.

Theorem 14: Let P be an arbitrary 5ot-cell partitioning of R^n Let f(x) be defined as x$^!$Mx, and let F(R) be a r^2 Lower-Bounding Function for f over S with positive constant

Q. Then for any iteration of the algorithm, the following upper bound on the number of active cells after that iteration holds:

$$|I(F,f^*)| \leq \frac{A}{C}(2D+\sqrt{D^2+QB^2})^n$$

Proof: By Lemma 18 in the Appendix, R is active only if *y* € E* where y = 8(D+>/D²+2S²). By Lemma 11, the upper bound of the number of active cells is

$$|I(F,f^*)| = |I(f(y(\cdot)),\gamma^2)| \leq \frac{A}{C}(2D+\sqrt{D^2+QB^2})^n$$

Q.E.D.

Hence, the number of active cells is bounded by the same constant for all iterations. The bounds for f* and P are given by

$$0 \leq f^* \leq \{\delta D\}^2$$

$$-Q\{\delta B\}^2 \leq F^* \leq 0$$

Hence, the difference between the two is bounded by

*f-F**< {8D}² + G{85}² = (D²+05²

which implies that the difference is quadratic in 5 or quartered for every iteration of the algorithm.

4. Location Examples

The generalized Weber problem that is examined here is also called the general min-sum location problem:

$$\text{GW)} \quad \min_{x \in R} \sum_{j=1}^{J} f_j(d(x, z_j))$$

where the Zj's are points in R^n, d(y) is the Euclidean metric, and each fj is a continuous non-decreasing function of the distance, d. If the fj are convex, GW can be shown to be convex and algorithms exist that are globally convergent for it. (See e.g. Katz [8] and Cooper [2].) Weaker restrictions on fj permit GW to have non-optimal local minima, hence the best that these algorithms can demonstrate is local convergence. (See e.g. [8].)

The feasible region is itself a rectangle, and is more of a restriction on the region to be searched derived perhaps from prior information rather than a set of requirements. For example, for many location problems, it can be shown that the optimal solution must lie in the convex hull of the locations of the fixed sources and sinks. Here, S could be the smallest rectangle containing this convex hull.

In this section, two lower-bounding functions for a single-facility location problem are examined with respect to their convergence rates. The first is the BSSS (Big Square-Small Square) algorithm given in Hansen et al [6]. It was constructed for use in constrained location problems and uses a simple lower-bounding function that has similarities to a Lipschitz lower-bounding function. A somewhat more complex (and accurate) lower-bounding function utilizing sub-functions is used in Edahl [4]. It demonstrated significantly better convergence than did the Hansen algorithm for the same unconstrained location problems.

Letting v be the optimal solution to GW, for the theorems of the previous section to be applied here, it is necessary that the objective function f(x) be able to approximated by a quadratic function in a neighborhood of v. This means that each of the fj(d(x,Zj)) be C^2 at v. Even if f^d) were C^2 for all positive d (as is the case for the power functions given below), it may not be differentiate at d=0 (as is the case for c<1 for the power functions below). However, dfx.z^ not being differentiate at x=Zj makes $f_J(d(x,z_j))$ differentiate only in unusual cases (i.e. c≥2 for the power functions given below). Hence for our analysis, we assume that v is not any of the z_i's. We also assume that none of the active cells contains a Zj.

4.1. BSSS Lower-Bounding Function

One of the simplest lower-bounding functions for GW is used in the BSSS (Big Square-Small Square) algorithm given in Hansen et al [6]:

$$F(R) = \sum_{j=1}^{J} f_j(d(w_j, z_j))$$

where *wj* is the solution to

$$\min_{w_j \in R} \ d(w_j, Zj) \qquad (1)$$

The computation of Wj is a relatively simple procedure, being the finding of the closest point in a cell to a given point z_{-y}

While not strictly a Lipschitz Lower-Bounding Function, it can be shown that F(R) can be bounded by two llbfs. Define L_j^{*m} and Lf^m by

$$L_j^{Max} = \underset{x \in S}{Max} fi\#x, \quad L?^{in} = \underset{x \in S}{Min} f_J<.d(x,z_J))$$

Further, let

$$p = \underset{i}{min} <xV5(a)$$

For any cell R, the following clearly hold:

$$jj(d(y(R), zp) - KKtf'' < fjWyp\, zp)$$

$$fj(d(y_{j9}\ zp) \leq S\ fj(d(y(R),\ zp) - r(R)\beta L_j^{Min}$$

Letting

$$L^{Max} = \sum_{j=1}^{J} L_j^{Max}; \quad L^{Min} = \beta \sum_{j=1}^{J} L_j^{Min}$$

we have

$$f[y(R)) - rL^{Max} <> F(R) \leq f(y(R)) - rL^{Min}$$

Hence F(R) is bounded from below by a Lipschitz lower-bounding function, and should have a convergence rate at least as good as that one. It is trivial to show that for any C^1 function f(x), any positive number is a Lipschitz number for f(x) for all x sufficiently close to a given local optima of f(x). Hence, for S sufficiently small and centered about the optima, the right hand side is a llbf for f(x), and hence bounds F(R) from above.

4.2. Quadratic Sub-Function

The idea here is to determine, for each cost function fj, a sub-function over a cell R that is of the form $a+bd^2$. Were all the fj's actually such quadratics, then the problem would be trivial to solve, for the problem would be a separable quadratic programming problem:

$$FQ(x) = \sum_{j=1}^{J} (<*j + bjd\{xap^2) = \sum_{j=1}^{J} (a_j + bfx-zp'ix-z_j))$$

$$= \sum_{j=1}^{J} a_j + x^t x \sum_{j=1}^{J} b_j - 2x^t \sum_{j=1}^{J} b_j z_j + \sum_{j=1}^{J} b_j z_j^t z_j$$

Were f not of this special quadratic form, the quadratic sub-function for each fj could be used. A lower bound for f in R could be gotten by minimizing the quadratic FQ over R. All that remains is to find the aj and bj coefficients (these depend of course on the cell R) to use in forming the quadratic function FQ(x).

4.2.1. Concave Cost Functions

Suppose that the **fj(d)** cost function were concave in d, besides being continuous and non-decreasing in d. While efficient algorithms exist for the case of fj(d) being <u>convex</u> in d, they are not guaranteed to be find the global optima should fj be concave.

Let Uj and Ij be given by (the subscripts on R are omitted here):

$$l_j = \min_{x \in R} d(x, Z_j); \quad U_j = \max_{x \in R} d(x, Z_j) \tag{2}$$

That is, these are the minimum and maximum distances from Zj to points in cell R. (One may use d(y(R),Zj)± r(R)) instead).

aj and bj are then given by:

$$a_j = \frac{u_j^2 f_j(l_j) - l_j^2 f_j(u_j)}{u_j^2 - l_j^2}; \quad b_j = \frac{f_j(u_j) - f_j(l_j)}{u_j^2 - l_j^2} \tag{3}$$

Note that b_j must be non-negative since fj is a non-decreasing function. Also note that

$$j \quad - \;^\frown + b_j u_j^2; \quad f_j(l_j) = a_j + b_j l_j^2$$

Hence this convex quadratic approximation must bound the concave fj from below for Ij < d < Uj. That is,

qfd) = aj + bjd^2 *<≥ffd)* **for** *ij <≤ d £ uj*

Consider a single fj(d) function. Let y be the center of cell R. For simplicity, suppose the subscripts j. Let d be any distance u>d>l. This will cover any point x in the cell R.

f is approximated by

K) = *M* +/(0(H) + /'(l)u(-0^2

Ad) =/(/) +/«X<H> +/'(5)(<H>^{2}

The quadratic sub-function for f over R is given by

$$q(d) = \frac{u^2 f(l) - l^2 f(u)}{u^2 - l^2} + \frac{f(u) - f(l)}{u^2 - l^2} d^2$$

$$= \frac{u^2 f(l \quad l^2 f(..)}{u^2 - l^2} \quad \frac{u) - f(l)}{u^2 - l^2}\{(d-l)^2 + 2(d-l)l + l^2\}$$

$$= f(l) + \frac{f(u) - f(l)}{u^2 - l^2}(d-l)(d+l)$$

Now, substituting for f(u) yields

= *f(l) + if(0(4-0 + f\r\)(u-t)(d-0){ 1 + ^}*
u+l

$$= f(l) + f'(l)(d-D \quad + f(f)(d-0 \frac{d-u}{u+l} + f''(\eta)(u-l)(d-l)\frac{d+l}{u+l}$$

Hence,

$$Ad) - q(d) = f'(\xi)(d-l)^2 - f(l)(d-l)\frac{d-u}{u+l} - f'(\eta)(u-l)(d-l)\frac{d+l}{u+l}$$

$$\leq \{f'(\xi) + \frac{f(l)}{u+l}\}(u-l)^2$$

Therefore, q(d) is an r^2 sub-function for f(d) over R. Since q and f are actually subscripted by j (for a single source or sink), it is f(x|R) given by

$$Ax\backslash R) = \sum_{j=1}^{J} q_j(d(x,z_j))$$

that is the r^2 sub-function of f over R.

4.2.2. Power Cost Functions

Suppose that each of fj(d) cost functions are of the form

$$ffd) = Wjtfh \quad Cj>0$$

This form is discussed is for example in [2] and [1], and allow for more accurate transportation cost fitting than the simple linear function (Cj=1).

There are three cases to be considered. The first is $l \geq Cj > 0$, the second is $2 \geq c_y > 1$, and the third is $c_j > 2$. In considering these cases, the subscript of j will be suppressed for notational simplicity. When the aj and bj terms have been computed, they can be used in forming FQ(x) to compute F(R).

For the case $l \geq c > 0$, f(d) would be concave, and the lower-bounding function from the previous section could be used. As shown in [4], the previous lower-bounding function could also be used for the $2 \geq c > 1$ case. In both cases, the coefficients a and b are given by:

$$a = w\frac{t f f R - i f_t}{ur^2 - r^2}; \; b = w\frac{i - f \; f}{ur^2 - r^2} \qquad (4)$$

This would then result in a r^2 lower-bounding function.

Now consider the case $c_y > 2$. As in [4], the lower-bounding function is constructed by making q(d) tangent to f at at point l>r>u. r is then selected to minimize the error at the endpoints (l and u). This results in the following lower-bounding function:

$$\frac{2}{c}\frac{u^c - l^c}{u^2 - l^2} = r^{c-2}$$

$$q(d) = w\frac{2-c}{2}r^c + w\frac{u^c - l^c}{u^2 - l^2}d^2$$

The difference between this formula for q(d) and the one for the other cases is simply the value of the constant term.

To show that q(d) is a r^2 lower-bounding function, let t be the designated tangent distance, l<t<u. By construction, a and b satisfy:

$a + bi^1 */®; \quad 2bt=f(t)$

$\rightarrow$

$a = f(t) - tf'(t)/2; \quad b = \frac{(t)}{2t}$

$q(d) = f(t) - tf'(t)/2 + \frac{f'(t)}{2t}d^2$

$= f(t) - tf'(t)/2 + \frac{f'(t)}{2t}\{(d-t)^2 + 2t(d-t) + t^2\}$

$= f(t) + f'(t)(d-t) + \frac{f'(t)}{2t}(d-t)^2$

Hence

$Ad) = / « +f(t)(d-t) + f''(\xi)(d-t)^2$

$f(d) - q(d) = \{f''(\xi) - \frac{f'(t)}{2t}\}(d-t)^2 \leq \{f''(\xi) \quad \frac{f'(t)}{2t}\}(u-l)^2$

5. Conclusion

Branch-and-bound algorithms may prove useful for the general global optimization problem. For the case of an unconstrained problem with a C^2 objective function, a lower-bounding function utilizing sub-functions (a i.e. r^2 lower-bounding functions) should give results far better than a Lipschitz lower-bounding function (or a related one such as that from the BSSS algorithm). It must be emphasized that these observations are for unconstrained problems. Should constraints be present or should the objective function be piecewise smooth, a Lipschitz lower-bounding function may give acceptable results.

I. Lemmas

The following lemmas are used in the proofs of the theorems in Section 3. For all of them, a fixed positive definite matrix M and positive cell vector a are assumed. Quantities such as A, B, C, and D, and functions f, g, r, I, and E are as defined in Sections 2 and 3.

Lemma 15: For given Y>0, let $P^{5,a}$ be an arbitrary 5oc-cell partitioning of R^n and I(x'y('y').γ*)[i] be the set of active cells. Let

$$\gamma^S = \gamma - \delta D; \ \gamma^L = \gamma + \delta D$$

Then

$$E^{\gamma^S} \subseteq E^{\gamma^L}$$

Proof: The two subset relations are proved separately:

a) Let x be any point in E^ and y_k be the center of any cell R_k containing x. If Y^S<0, then E^ will be empty, and the left subset relation satisfied trivially. Consider the case where ^>*Q.* The following hold:

[yk'Myk}^m ≤{(yrxyM(yrx))^irl +{x?Mx)^m **(By the triangle inequality)**

{(x-ykyM(x-yk)}^xri ≤g[R(0,S)) = 5D *(xe Rk)*

*{xfMx^Z^* *(xeF?)*

Substituting the second and third of these relations into the first give:

*[yjlMyk]^m ≤{(yrxyM(yk-x)}^1/2 +{*Mx)^1/2* < 5D + / = Y

Therefore *ke* /(/(yQXY²) and hence) xe 7^.

b) Let x be an arbitrary point in T^z and y_k be the center of any active cell containing x. Arguing in a similar way as in part a,

$$\{x^tMx\}^{1/2} \leq \{(x-y_k)^tM(x-y_k)\}^{1/2} + \{y_k{}^tMy_{lc}\}^{1/2} \leq 5D + Y = Y^{\wedge}$$

Therefore, *xfMx* ≤ *{+}²*. Hence *xe E^*. Q.E.D.

Lemma 16: Let Y^M>0 and L>Y^ML° be given. Let $P^{5,a}$ be an arbitrary Sa-ceii partitioning of R^n with

5<Jf }²

Let Ysatisfy Y^2 ^ 6SL + $\{5D\}^2$. Then

1) L is a valid Lipschitz number for the function $f(x)=x^tMx$ over the ellipse E^{γ^M}.

and

2) $\gamma^L = \gamma + \delta D \leq \gamma^M$

Proof: The first part follows directly from the definition of Lipschitz number.

---

[1] R_p is in I if $f(y(R_p))=f(y_p) \leq \gamma^2$

2) First we show that 8D < f^1

$$\delta < \frac{(\gamma^M)^2}{BL+2\gamma^M D} < \frac{(\gamma^M)^2}{2\gamma^M D} = \frac{\gamma^M}{2D}$$

Now,

$$\delta \leq \frac{(\gamma^M)^2}{BL+2\gamma^M D}$$

$\rightarrow$

$$\delta BL + 2\delta\gamma^M D \leq (\gamma^M)^2$$

$\rightarrow$

$$\delta BL + (\delta D)^2 \leq (\gamma^M)^2 - 2\delta\gamma^M D + (\delta D)^2$$

$\rightarrow$

$$\sqrt{\delta BL + (\delta D)^2} \leq \gamma^M - \delta D$$

$\rightarrow$

$$\gamma^L = \gamma + \delta D \leq \sqrt{\delta BL + (\delta D)^2} + \delta D \leq \gamma$$

Q.E.D.

Lemma 17: Let f(x) be defined as $x^t Mx$, and let F(R) be a Gradient Lower-Bounding Function for R with positive constant K. Then R is an active cell only if

$$\sqrt{y^t My} \leq \delta(KBL^0+D)$$

where y is the center of R.

Proof: R is active only if $F(R) \leq f^* \leq (\delta D)^2$, where f* is the value of the best point found so far. By the definition of glbf, F(R) must also satisfy

$$f(y) - \delta KB\overline{L}(R) \leq F(R)$$

therefore,

$$f(y) - \delta KB\overline{L}(R) \leq F(R) \leq (\delta D)^2$$

Now, since R must be in the ellipse $E^{\sqrt{y^t My}+\delta D}$,

$$\overline{L}(R) \leq L^0(\sqrt{y^t My}+\delta D)$$

Therefore, the following must hold for R to be active:

$$f(y) - \delta KBL^0(\sqrt{y^t My}+\delta D) \leq (\delta D)^2$$

Rearranging terms,

$$y^t My - \delta\sqrt{y^t My}(KBL^0) - \delta(DKBL^0+D^2) =$$
$$(\sqrt{y^t My} + \delta D)(\sqrt{y^t My} - \delta(KBL^0+D)) \leq 0$$

Since the first term is always positive, R can be active only if

$$\sqrt{y^t My} \leq \delta(KBL^0+D)$$

Equivalently, $y \in E^\gamma$ where $\gamma = \delta(KBL^0+D)$. Q.E.D.

Lemma 18: Let f(x) be defined as $x^!Mx$, and let F(R) be a r^2 Lower-Bounding Function for R with positive constant Q. Then R is an active cell only if

$$\sqrt{y^tMy} \le \delta(D+\sqrt{D^2+QB^2})$$

where y is the center of R.

Proof: R is active only if *F(R)*≾/ ≤ $\{5D\}^2$ where f* is the value of the best point found so far. For any x in R, the following holds true by the triangle inequality:

$$\sqrt{y^tMy} \le \sqrt{x^tMx} + <\overline{(x-yyM(x-y)}$$

$$\rightarrow$$

$$\sqrt{f(y)} \le \sqrt{f(x)} + \delta D$$

$$\rightarrow$$

$$\min_{x \in R} \sqrt{f(x)} \ge \sqrt{f(y)} - \delta D$$

By the definition of an r^2lb function,

Ax\R)≥Ax)-Q>iR)^2 VJCG/?

$$F(R) = \min_{x\in R} fWR) * \min_{xeR} f\{x)-Qr(R)^2 \ge \{\sqrt{f(y)} - \delta D\}^2 - Q\{\delta B\}^2$$

Hence, R is active only if

\B)^2≾*f* <≥ *{8D}*^{2}

→

fly) — l&D^lfty) + {8D}^{2} — *Q*!{85}^{2} £ {5D}^{2}

→

J(y)—28zW/(y)—*Q{SB}*^{2} =

Since the second term is always positive, we have R is active only if

$$\sqrt{y^tMy} = \sqrt{f(y)} \le \delta(D + \sqrt{D^2+QB^2})$$

Q.E.D.

REFERENCES

[1] Chen, Reuven.
Solution of Location Problems with Radial Cost Functions.
Computers and Mathematics with Applications 10(1):87-94,1984.

[2] Cooper, Leon.
An Extension to the Generalized Weber Problem.
Journal of Regional Science 8() :181 -197,1968.

[3] Dixon, L.C.W. and Szego, G.P.
Towards Global Optimisation 2.
North-Holland Publishing Company, New York, 1978.

[4] Edahl, R.H.
A Spatial Branch-and-Bound Algorithm for Some Unconstrained Single-Facility Location Problems.
EDRC Technical Report ():, 1989.

[5] Goldberg, R.R.
Methods of Real Analysis.
Blaisdell Publishing Company, Waltham, Massachusetts, 1964.

[6] Hansen, P, Peeters, D., Richard, D, and Thisse, J.
The Minusum and Minimax Location Problems Revisited.
Operations Research 33(6):1251-1265,1985.

[7] Horst, R.
A General Class of Branch-and-Bound Methods in Global Optimization with Some New Approaches for Concave Minimization.
Journal of Optimization Theory and Applications 51 (2) :271 -291,1986.

[8] Katz, I.N.
Local Convergence in Fermat's Problem.
Mathematical Programming 6():89-104,1974.

[9] McCormick, G.P.
Computability of Global Solutions to Factorable Nonconvex Programs.
Mathematical Programming 10():147-175,1976.

[10] Meewella, C. C, and Mayne, D. Q.
An Algorithm for Global Optimization of Lipschitz Continuous Functions.
Journal of Optimization Theory and Applications 57(2) :307-322, May, 1988.

[11] Mitten, I.G.
Branch-and-Bound Methods: General Formulations and Properties.
Operations Research 8():24-34,1970.

[12] Shubert, Bruno, O.
A Sequential Method Seeking the Global Maximum of a Function.
SIAM Journal of Numerical Analysis 9(3):379-388,1972.