LP-Form Inclusion Functions for Global Optimization

Stuart G. Mentzer
Objexx Engineering
Natick MA 01760 USA

Based on an article in *Computers Math. Applic.* Vol 21, No. 6/7, pp. 51-65, 1991

Abstract. A new class of methods is presented for finding the global extrema of real-valued functions using point values and gradient inclusions. These methods construct polyhedral envelopes for each subfunction within regions generated by a subdivision strategy. The range of the envelopes determines bounds on the functions and can be computed by linear programs. Partial monotonicity conditions are presented that reduce the effective dimension or yield exact bounds on finite regions.

1 Introduction

Global optimization is concerned with finding extrema of a general class of real-valued functions when no simplifying global properties such as convexity are known. A global minimization problem can be stated as simply as

$$\min \; z(\boldsymbol{x}) : \mathbf{R}^n \to \mathbf{R} \qquad \text{for} \quad \boldsymbol{x} \in D \subseteq \mathbf{R}^n, \;\; D \text{ compact.}$$

The dimension of this problem is the number of variables, n. The NP-hardness of this class of problems makes worst-case polynomial time algorithms unlikely. Efforts in global optimization have instead focused on stochastic methods, which assure some specific localization of the optimal value with probability approaching one [2, 3, 16], or deterministic methods, which guarantee asymptotic convergence to the optima but with a worst-case rate of convergence that is exponentially slow in the problem dimension. Deterministic methods tailored to separable, concave, and other special problem classes have better performance [4, 17].

Our focus is on tools for deterministic methods that apply to a broad class of nonconvex, nonlinear programming problems. We require only that the objective $z(\boldsymbol{x})$ and the constraints defining D be factorable C^1 functions. A number of branch and bound methods have been developed for these problems that can be shown to converge to a superset of global optima and to the optimal value under fairly weak conditions [6, 7, 10, 13, 14, 15]. With each of these methods, the branching, or subdivision, strategies generates a list of subregions of D that are still candidates to contain a global optimum. Each region is evaluated for bounds on the objective function value and feasibility of the constraints. Tests for the existence of stationary points, convexity, or other necessary and sufficient optimality conditions have also been used. All of these methods requires bounds on the objective, constraints, and their gradients and Hessians. Regions shown not to contain the global optimum can be discarded, but the remaining regions must be subdivided to obtain refined bounds.

Each bounding problem occurring in these global optimization methods is an instance of the generic problem: *Find an inclusion for the range of a real-valued function $z(\boldsymbol{x})$ on a compact region $D \in \mathbf{R}^n$.* This is the problem we address.

Deterministic approaches have typically used fairly simple bounding techniques, depending heavily on the subdivision process. In problems of even moderate dimension the expense of more accurate bounds can be more than justified against the exponential cost associated with subdivision. In particular, we focus here on the "early" phase where the number of regions can quickly proliferate unless exceptionally accurate bounds are used.

We present a new approach to the bounding problem. For a given subregion under consideration, we construct a polyhedral envelope for each subfunction of $z(\boldsymbol{x})$ based on function values and bounds on the gradient. Given this information we show how to compute (in some cases optimal) bounds using linear programs for the lower and upper envelopes. The resulting range inclusions are better than the well-known natural interval extension and mean value form of interval analysis. We also present a procedure for detecting and exploiting partial monotonicities within this framework that can successively reduce the effective dimension of subregions, and which produces the exact range on regions with a weak monotonicity property. Thus the combined approach not only provides strong bounds, but also attacks the inherent "curse of dimensionality".

1.1 Interval Analysis

Interval analysis allows the construction of numerical bounds on real-valued factorable functions over finite boxes. Let $\mathbf{I}$ be the set of real compact intervals $[a, b]$ with $a, b \in \mathbf{R}$. If $X = [a, b]$, we denote the lower and upper bounds of X by $\underline{X} = a$ and $\overline{X} = b$. The *width* of an interval is defined by $w(X) = \overline{X} - \underline{X}$. We define the set of n-dimensional boxes, $\mathbf{I}^n$, to be the set of Cartesian products of n intervals, such as

$$\boldsymbol{X} \;=\; (X_1, \ldots, X_n) \;=\; [a_1, b_1] \times \cdots \times [a_n, b_n].$$

The width of a box is defined by $w(\boldsymbol{X}) = \max_i (w(X_i))$.

Operations in $\mathbf{I}$ are set-valued extensions of operations over the reals, so for $* \in \{+, -, \times, \div\}$

$$X * Y \;\doteq\; \{x * y : \; x \in X, \; y \in Y\}.$$

Properties of interval operations are discussed in [1, 13, 14, 15].

Factorable functions [10] are simply functions $z(\boldsymbol{x})$ that can be represented as a rooted directed acyclic graph with the variables at the in-degree zero leaf nodes and $z(\boldsymbol{x})$ at the root. Each nonleaf node represents a sum or product of its subfunctions or a unary function of a single subfunction. Since taking an inverse is a unary function, quotients are clearly allowable operations. Similarly, an exponentiation operation $y(\boldsymbol{x})^{r(\boldsymbol{x})}$ can be converted to $\exp[r(\boldsymbol{x})\ \ln(y(\boldsymbol{x}))]$.

For a real-valued factorable function $f(\boldsymbol{x})$, we denote the lower and upper bounds of $f(\boldsymbol{x})$ on a box $\boldsymbol{X}$ by

$$\underline{f}(\boldsymbol{X}) \doteq \{\inf\left(f(\boldsymbol{x}) : \boldsymbol{x} \in \boldsymbol{X}\right)\} \qquad \text{and} \qquad \overline{f}(\boldsymbol{X}) \doteq \{\sup\left(f(\boldsymbol{x}) : \boldsymbol{x} \in \boldsymbol{X}\right)\}$$

and the range of $f(\boldsymbol{x})$ by $f(\boldsymbol{X}) \doteq \left[\underline{f}(\boldsymbol{X}), \overline{f}(\boldsymbol{X})\right]$.

We call $F : \mathbf{I}^n \to \mathbf{I}$ an *inclusion* for $f(\boldsymbol{x})$ on the box $\boldsymbol{X}$ if $F(\boldsymbol{Y}) \supseteq f(\boldsymbol{Y})$ for all $\boldsymbol{Y} \subseteq \boldsymbol{X}$. We call the restricted inclusion $F : \mathbf{R}^n \to \mathbf{I}$ an *inclusion function*. An inclusion $F : \mathbf{I}^n \to \mathbf{I}$ is called *Lipschitz* [13] if there exists a $K \in \mathbf{R}$ such that

$$w(F(\boldsymbol{X})) \leq K w(\boldsymbol{X}) \quad \forall \boldsymbol{X} \in \mathbf{I}^n.$$

An inclusion $F :\ \mathbf{I}^n \to \mathbf{I}$ of a function $f :\ \mathbf{R}^n \to \mathbf{R}$ is said to have *convergence of order* α [15] if for $\boldsymbol{X} \in \mathbf{I}^n$

$$w(F(\boldsymbol{X})) - w(f(\boldsymbol{X})) = \mathcal{O}(w(\boldsymbol{X})^\alpha).$$

An inclusion $F :\ \mathbf{I}^n \to \mathbf{I}$ of a function $f :\ \mathbf{R}^n \to \mathbf{R}$ is said to be *inclusion monotonic* if

$$\boldsymbol{X} \subseteq \boldsymbol{Y} \quad \Rightarrow \quad F(\boldsymbol{X}) \subseteq F(\boldsymbol{Y}).$$

We restrict factorable functions to allow only those unary functions for which inclusion monotonic inclusions are readily computed. In practice we can compute inclusions of arbitrary precision for common unary functions such as $\sin(\cdot)$, $\ln(\cdot)$, or $(\cdot)^2$.

Interval analysis provides a simple procedure for computing an inclusion of the range of a factorable function. The *natural interval extension* of a given representation of a real-valued factorable function $f(\boldsymbol{x}) : \mathbf{R}^n \to \mathbf{R}$ on a box $\boldsymbol{X} \in \mathbf{I}^n$ is the inclusion $N(\boldsymbol{X})$ obtained by replacing all occurrences of each variable x_i with the corresponding interval X_i, each unary subfunction $u(\cdot)$ with its inclusion $U(\cdot)$, and each operation with the corresponding interval operation [11]. The definition of the interval operations guarantees that $N(\boldsymbol{X}) \supseteq f(\boldsymbol{X})$.

The natural interval extension is, in general, strictly larger than the actual function range because the intervals allow inconsistent values to be used for variables appearing in two or more subfunctions. Nevertheless, it is not hard to see that the natural interval extension is the tightest inclusion possible when each function receives only the inclusion interval of its subfunctions for use in computing its inclusion. The natural interval extension of most continuous, bounded factorable functions is Lipschitz and therefore linearly convergent [14].

For simplicity, we assume exact interval arithmetic. These results are readily modified to the case of machine interval arithmetic. We also assume the extended form of interval arithmetic that can handle unbounded intervals, as defined in [15].

1.2 Interval Inclusion Methods

A variety of methods for finding an inclusion of a function using interval analysis have been applied to global optimization [6, 15]. The most popular inclusions that apply to a broad class of functions $f(\boldsymbol{x})$ are

$$\begin{aligned} & N(\boldsymbol{X}) && \text{(natural interval extension)} \\ T_1(\boldsymbol{X}) &= f(\boldsymbol{c}) + \boldsymbol{G}(\boldsymbol{X})(\boldsymbol{X}-\boldsymbol{c}) && \text{(mean value form)} \\ T_2(\boldsymbol{X}) &= f(\boldsymbol{c}) + \boldsymbol{\nabla} f(\boldsymbol{c})(\boldsymbol{X}-\boldsymbol{c}) + \tfrac{1}{2}(\boldsymbol{X}-\boldsymbol{c})^T \mathbf{H}(\boldsymbol{X})(\boldsymbol{X}-\boldsymbol{c}) && \text{(Taylor form of order 2)} \end{aligned}$$

where $\boldsymbol{c} \in \boldsymbol{X}$, $\boldsymbol{\nabla} f$ is the row vector gradient, $\boldsymbol{G}(\boldsymbol{X})$ is an inclusion for $\boldsymbol{\nabla} f(\boldsymbol{X})$, and $\mathbf{H}(\boldsymbol{X})$ is an inclusion for the Hessian. The mean value form requires $f(\boldsymbol{x}) \in C^1$ and the Taylor form of order 2 requires $f(\boldsymbol{x}) \in C^2$. There are higher-order Taylor forms (the mean value form is the Taylor form of order 1) but these are rarely used. Just as the $N(\boldsymbol{X})$ is the interval version of a function evaluation, the Taylor forms are interval versions of Taylor's theorem. Interval adaptations of Newton's method have also been used to assess the existence of stationary points [6].

The convergence properties of these inclusions have been investigated [15]. The natural interval extension is linearly convergent if it is Lipschitz. The mean value form is quadratically convergent if $\boldsymbol{G}(\cdot)$ is Lipschitz. The Taylor form of order 2 is quadratically convergent if $\mathbf{H}(\cdot)$ is bounded.

These bounding tools are used with subdivision strategies to localize the global optima. Typically, the regions considered are boxes and subdivision of a box involves partitioning the box in half along one of the coordinate directions. If we have an inclusion $F(\boldsymbol{Y}) \supseteq f(\boldsymbol{Y})$ that is valid for all $\boldsymbol{Y} \subseteq \boldsymbol{X}$, and a partition $\boldsymbol{X} = \bigcup_j \boldsymbol{X}_j$, we take $F(\boldsymbol{X}) = \bigcup_j F(\boldsymbol{X}_j)$ as the inclusion on $\boldsymbol{X}$. As the subboxes get smaller, these inclusions converge, and at some point we can eliminate infeasible, nonoptimal, or suboptimal regions. The global convergence behavior of some basic subdivision strategies is reviewed in [15].

The combination of a quadratically convergent inclusion and a globally convergent subdivision strategy sounds more powerful than it is. The convergence is quadratic in the width of a box, but it takes a partition into $\mathcal{O}(k^n)$ subboxes to achieve a factor of k reduction in the width in an n-dimensional problem. Ratschek and Rokne [15] have shown that for the problem of computing the range of a function $f(\boldsymbol{x})$ on an initial box $\boldsymbol{X}$, given an inclusion $F(\cdot)$ that has convergence of order α and is inclusion monotonic, the worst-case convergence is given by

$$w(F(\boldsymbol{X})) - w(f(\boldsymbol{X})) = \mathcal{O}(\ell^{-\alpha/n})$$

where ℓ is the number of boxes examined. Thus the number of boxes may explode before a significant part of the initial region has been eliminated. We cannot hope to solve high-dimension problems by depending on subdivision to improve the bounds of weak inclusions.

The inclusions given above, although readily computed, do not make much effort to generate strong bounds. Each of these uses a natural interval extension for the function, gradient, or Hessian of the full function. No attempt is made to find the best inclusion given all of the subfunction information.

1.3 Nested Inclusion Functions

To generate better inclusions we need to construct inclusion functions for each subfunction that can be propagated through the function graph into an inclusion function for $z(\boldsymbol{x})$ which is amenable to optimization.

The interval analysis methods have implicit inclusion functions that could be used in this way. For example, the mean value form inclusion function is

$$T_1(\boldsymbol{x}) = f(\boldsymbol{c}) + \boldsymbol{G}(\boldsymbol{X})(\boldsymbol{x} - \boldsymbol{c}).$$

We will see that the nonconvexity of this form makes it difficult to optimize and propagate.

A natural choice is to use convex envelope inclusion functions. The *envelope* of an inclusion function $F(\boldsymbol{x})$ on $\boldsymbol{X}$ is defined as

$$S(\boldsymbol{X}, F) = \{(\boldsymbol{x}, y) : \boldsymbol{x} \in \boldsymbol{X},\ y \in F(\boldsymbol{x})\}.$$

McCormick [10] has developed a method for constructing convex envelope inclusion functions. Some slack is introduced in this process, so the resulting inclusion function for $z(\boldsymbol{x})$ is generally larger than the actual convex envelope. The intermediate steps involve some complex computations and the envelope representation can grow quite large and complicated. Nevertheless, this is an elegant approach that can produce inclusions of high quality.

We take a middle approach, propagating polyhedral inclusion functions that are amenable to solution by linear programming. Like McCormick's method, this is a purely global tool, not an adaptation of a local method. In the basic approach, we simplify the inclusion functions for each subfunction before propagating them up to the next level. This prevents the resulting polyhedral envelopes from becoming too complicated and simplifies the presentation. In Section 4 we indicate how to generalize this to use the full envelopes. We begin by examining the choices for inclusion of a univariate function.

2 Univariate LP-Forms

We consider the problem of finding bounds on a factorable univariate function $z(x) \in C^1$ on the interval $X \in \mathbf{I}$, given a representation of $z(x)$ in terms of subfunctions. For each function we compute an inclusion function from the inclusion functions of its subfunctions. Here we assume that this inclusion function will be simplified to depend only on an overall inclusion range and derivative inclusion. Thus we are interested in how good an inclusion range can be computed from the inclusion functions of subfunctions. First we look at a function that is a sum of subfunctions.

2.1 Sums

Consider a function $f(x) = \sum_j f_j(x)$ where the $f_j(x)$ are C^1 and where the inclusions $F_j(X) \supseteq f_j(X)$ and $F'_j(X) \supseteq f'_j(X)$ are known and we are given the function values $f_j(c)$ for some $c \in X$. We wish to compute an inclusion $F(X)$ for $f(X)$.

The natural interval extension gives the inclusion

$$N(X) = \sum_j F_j(X)$$

and the mean value form gives the inclusion

$$T_1(X) = \sum_j f_j(c) + \left(\sum_j F'_j(X) \right) (X - c)$$

for any $c \in X$ selected. It is not difficult to construct the optimal inclusion function $F^*(x)$ for the same information.

Lemma 1 *For a function $f(x) = \sum_j f_j(x)$, with all $f_j(x) \in C^1$, given only the inclusions $F_j(X) \supseteq f_j(X)$ and $F'_j(X) \supseteq f'_j(X)$ and the values $f_j(c)$ for $c \in X \in \mathbf{I}$ and for all j, the inclusion function for $f(x)$*

$$F^*(x) = \sum_j F^*_j(x) = \sum_j \left[F_j(X) \bigcap \left(f_j(c) + F'_j(X)(x - c) \right) \right] \tag{2.1}$$

is optimal. That is, for any other inclusion function $E(x) \supseteq f(x)$, $F^*(x) \subseteq E(x)$ for all $x \in X$.

Proof. First we show that the $F^*_j(x)$ are optimal inclusions of the $f_j(x)$ for the given information. Suppose $E_j(x)$ is a better inclusion function of $f_j(x)$ on X. For any $x \in X$ where $F^*_j(x) \not\subseteq E_j(x)$, a C^1 function $\hat{f}_j(x)$ can be constructed that satisfies

$$\hat{f}_j(c) = f_j(c)\ , \quad \hat{f}_j(X) \subseteq F_j(X)\ , \quad \text{and} \quad \hat{f}'_j(X) \subseteq F'_j(X)$$

but for which $\hat{f}_j(x) \notin E_j(x)$, contradicting the claim that $E_j(x)$ is an inclusion function for any C_1 function $f_j(x)$ for which $F_j(X)$, $F'_j(X)$, and $f_j(c)$ are valid. Such a function $\hat{f}_j(x)$ can be pasted together from at most five function segments, including one sloped linear function, two circular arcs, and two constant functions.

Given no other information about the functions $f_j(x)$, the $F^*_j(x)$ are independent and the optimal inclusion function for $f(x)$ with $x, c \in X$ is simply the sum $F^*(x)$ given in the lemma. □

It is not hard to see that $|y-c| < |x-c| \Rightarrow F^*(y) \subseteq F^*(x)$, thus the optimal inclusion is

$$F^*(X) = F^*(\underline{X}) \bigcup F^*(\overline{X})$$

which is readily computed. The ranges $N(X)$ or $T_1(X)$ are obtained by ignoring one or the other of the intersected ranges, verifying their nonoptimality. The inclusion $F^*(X)$ has the desirable properties of these weaker inclusions.

Lemma 2 *If the inclusions $F_j(X)$ in (2.1) are Lipschitz then the inclusion $F^*(X)$ is linearly convergent. If the inclusions $F'_j(X)$ are Lipschitz then the inclusion $F^*(X)$ is quadratically convergent.*

This lemma follows directly from the respective properties of $N(X)$ and $T_1(X)$ [14], and the fact that $w(F^*(X)) \leq \min(w(N(X)),\ w(T_1(X)))$ which follows from Lemma 1.

2.2 LP-Forms for Sums

It is not difficult to extend the above approach to obtain optimal inclusions when other information is provided on some interval. One drawback to using interior values $f_j(c)$ is that the resulting inclusion function envelope,

$$S(X, F^*) = \{(x, y) : x \in X,\ y \in F^*(x)\},$$

is not in general convex. Thus finding $F^*(X)$ would be difficult in higher dimensions. Another drawback is that $F^*(X)$ does not generally produce exact bounds even when $f(x)$ is known to be monotonic on X, for example when $\sum_j F'_j(X)$ is nonnegative or nonpositive. An alternative which avoids these drawbacks is to use the function values $f_j(\underline{X})$ and $f_j(\overline{X})$ instead of $f_j(c)$ in the inclusion.

Lemma 3 *For a function $f(x) = \sum_j f_j(x)$, with all $f_j(x) \in C^1$, given only the inclusions $F_j(X) \supseteq f_j(X)$ and $F'_j(X) \supseteq f'_j(X)$ and the values $f_j(\underline{X})$ and $f_j(\overline{X})$ for all j, the inclusion function for $f(x)$*

$$\begin{aligned} F^\diamond(x) &= \textstyle\sum_j F_j^\diamond(x) \\ &= \textstyle\sum_j \left[F_j(X) \bigcap \left(f_j(\underline{X}) + F'_j(X)(x - \underline{X}) \right) \bigcap \left(f_j(\overline{X}) + F'_j(X)(x - \overline{X}) \right) \right] \end{aligned}$$

is optimal. That is, for any other inclusion function $E(x) \supseteq f(x)$, $F^\diamond(x) \subseteq E(x)$ for all $x \in X$.

The proof of this lemma is analogous to that of Lemma 1, and $F^\diamond(x)$ also shares the convergence properties of $F^*(x)$ given in Lemma 2. It is not difficult to show that the envelope of $F^\diamond(x)$ is convex on X. Since $F^\diamond(x)$ is also polyhedral, we call it an LP-form.

Definition. An inclusion function $F(\boldsymbol{x})$ is an *LP-form* on $\boldsymbol{X}$ if the envelope of $F(\boldsymbol{x})$ on $\boldsymbol{X}$ is polyhedral (and thus convex).

The inclusion range implied by an LP-form can be computed by solving two linear programs whose constraint sets are precisely the lower and upper envelopes of the LP-form. We call these the primal linear programs. Some insight into the significance of monotonicity in the LP-form can be gained by considering the dual problem of finding the best bounds on the inclusion range.

For the subfunction $f(x) = \sum_j f_j(x)$ where the $f_j(x)$ are C^1, consider the representation

$$f(x) \;=\; \left(\sum_j \alpha_j f_j(x) + \delta x\right) \;+\; \left(\sum_j \beta_j f_j(x) - \delta x\right) \;+\; \left(\sum_j \gamma_j f_j(x)\right) \tag{2.2}$$

where $\alpha_j, \beta_j, \gamma_j, \delta \in \mathbf{R}$ and $\alpha_j + \beta_j + \gamma_j = 1$ for all j.

We can construct linear programs for upper and lower bounds on $f(X)$ based on this representation and using only the values $f_j(\underline{X})$, $f_j(\overline{X})$, $F_j(X)$, and $F_j'(X)$. The idea will be to find the partition into monotonically increasing, monotonically decreasing, and non-monotonic terms which gives the best bounds. The linear program for the lower bound is

$$(\mathrm{D}^{\Sigma}) \qquad \max \; \left(\sum_j \alpha_j f_j(\underline{X}) + \delta \underline{X}\right) \;+\; \left(\sum_j \beta_j f_j(\overline{X}) - \delta \overline{X}\right) \;+\; \left(\sum_j \gamma_j \underline{F}_j(X)\right)$$

subject to

$$\begin{array}{rcll}
\sum_j \alpha_j \underline{F}_j'(X) \,+\, \delta & \geq & 0 & \\
\sum_j \beta_j \overline{F}_j'(X) \,-\, \delta & \leq & 0 & \\
\alpha_j \,+\, \beta_j \,+\, \gamma_j & = & 1 & \forall\, j \\
\alpha_j,\ \beta_j,\ \gamma_j & \geq & 0 & \forall\, j
\end{array}$$

The first constraint assures that the first grouped term in (2.2) is monotone nondecreasing and the second constraint assures that the second term is monotone nonincreasing. The objective terms are valid lower bounds given their monotonicities. The last term is the non-monotonic part for which the corresponding objective term is clearly a valid lower bound. Any feasible solution to D^{Σ} is then a valid lower bound on $f(X)$. More importantly, an optimal solution to D^{Σ} gives the optimal lower bound $\underline{F}^{\diamond}(X)$.

Theorem 4 *The optimal value of the linear program D^{Σ} is the optimal lower bound $\underline{F}^{\diamond}(X)$ of $f(x) = \sum_j f_j(x)$ for $x \in X$ given only the values $f_j(\underline{X})$, $f_j(\overline{X})$, $F_j(X)$, and $F_j'(X)$ for all j.*

Proof. The primal linear program for $\underline{F}^{\diamond}(X)$ is

$$(\mathrm{P}^{\Sigma}) \qquad \min \sum_j e_j$$

subject to

$$\begin{aligned}
e_j &\geq f_j(\underline{X}) + \underline{F}'_j(X)(x - \underline{X}) && \forall\, j \\
e_j &\geq f_j(\overline{X}) + \overline{F}'_j(X)(x - \overline{X}) && \forall\, j \\
e_j &\geq \underline{F}_j(X) && \forall\, j \\
x &\geq \underline{X} \\
x &\leq \overline{X}
\end{aligned}$$

It is not hard to show that P^{Σ} is indeed dual to D^{Σ}; thus they both produce the optimal value $\underline{F}^{\diamond}(X)$. □

The optimal upper bound $\overline{F}^{\diamond}(X)$ can be found by analogous linear programs. Note that the optimal e_j values will be $\underline{F}^{\diamond}_j(x)$ for the optimal x value.

The feasible solutions to the dual D^{Σ} are worth examining. Setting $\gamma_j = 1$ for all j and all other variables to zero reproduces the natural interval extension bound $N(X)$. Setting all the $\alpha_j = 1$ or all the $\beta_j = 1$ and taking the least feasible δ value gives the $T_1(X)$ bounds expanded around $\underline{X}$ or $\overline{X}$, respectively. When $f(x)$ is shown to be monotonic, that is, when $\sum_j F'_j(X)$ is nonnegative or nonpositive, D^{Σ} has as a feasible solution one of the boundary values $f(\underline{X})$ or $f(\overline{X})$. The monotonicity proves that this is also the exact bound of $f(X)$ and thus it is an optimal solution to D^{Σ}. This assures that an LP-form inclusion with convergent (of any order) inclusions $F'_j(X)$ converges to exact bounds on any region of strict monotonicity in a finite number of subdivision steps. We note that the LP-form inclusion is quadratically convergent if either $\underline{F}'_j(\cdot)$ or $\overline{F}'_j(\cdot)$ is Lipschitz, since it lies inside the mean value forms with $c = \underline{X}$ and $c = \overline{X}$.

By comparing the number of variables and the number of constraints, we see that at most four of the α_j, β_j, or γ_j can be nonintegral at an optimal feasible solution; thus at most two of the f_j are fractionally decomposed. The dual can be simplified in a number of ways. For example, if $\underline{F}'_j(X) \geq 0$ or $\overline{F}'_j(X) \leq 0$, we can set $\gamma_j = 0$. The solution of the primal can be efficiently found by sorting the set of breakpoints in the $\underline{F}_j(x)$ functions and the endpoints $\underline{X}$ and $\overline{X}$, and performing a discrete Fibonacci search [9] for the minimum value of the convex objective $\sum_j e_j$.

Example

Although LP-form inclusions are intended to extend the class of higher-dimension, nested factorable functions that can be optimized, the quality of the LP-form bounds can be illustrated with a simple one-dimensional sum.

Consider the problem of finding a lower bound on the function

$$f(x) \;=\; 2x^2 - 4x^{3/2} + 5\sin(7x)$$

on $[0,5]$. This function has five interior local minima, with the minimum

$$\underline{f}(X) \;\approx\; f(2.4666) \;\approx\; -8.3270.$$

We consider lower bounds for three successively smaller intervals containing the minimum point. The natural interval extension bound is given by

$$\underline{N}(X) \;=\; 2\,\underline{X}^2 - 4\,\overline{X}^{3/2} + 5\,\underline{\sin(7X)}.$$

The mean value form using the natural interval extension $F'(X) \supseteq f'(X)$ is given by

$$\underline{T}_1(X) \;=\; f(c) + \underline{F'(X)(X-c)}$$

where we can choose c to give the best bound from

$$c = \begin{cases} \underline{X} & \text{if} \quad \underline{F}' \geq 0 \\ \overline{X} & \text{if} \quad \overline{F}' \leq 0 \\ \dfrac{\overline{F}'(X)\underline{X} - \underline{F}'(X)\overline{X}}{\overline{F}'(X) - \underline{F}'(X)} & \text{otherwise.} \end{cases}$$

The Taylor form of order 2 using the natural interval extension $F''(X) \supseteq f''(X)$ is given by

$$\underline{T}_2(X) \;=\; f(c) + f'(c)(X-c) + \tfrac{1}{2}\underline{F''(X)(X-c)^2}$$

where c is the center point of X (the best point can not be determined from $F''(X)$).

For $X = [0,5]$ we get

$$\underline{N}(X) \;=\; 0 - 4 \cdot 5^{3/2} - 5 \;\approx\; -49.72$$

$$\underline{T}_1(X) \;\approx\; f(2.3408) + (-48.4164)(5 - 2.3408) \;\approx\; -135.25$$

$$\underline{T}_2(X) \;=\; -\infty$$

and the LP-form lower bound (shown as the optimal dual objective) is

$$\underline{F}^{\diamond}(X) \;\approx\; (.329 \cdot 0 + 1 \cdot 0 + 13.416 \cdot 0) + (.671 \cdot 50 - 13.416 \cdot 5) + (-5) \;\approx\; -38.54.$$

The table below summarizes the results for all three intervals.

On $X = [2.4, 2.6]$, the second derivative is provably positive from $F''(X)$, and in practice a more efficient local search method could be used to complete the solution.

The additional dependence on subdivision of the standard methods can be seen even in this simple one-dimensional example. For example, to obtain T_1 bounds that at least match the LP-form bounds requires examining at least seven boxes for each of the three starting intervals. The cost of even this amount of finer granularity in a higher-dimension problem would be dramatic.

X	$\underline{N}(X)$	$\underline{T}_1(X)$	$\underline{T}_2(X)$	$\underline{F}^\diamond(X)$
$[0,5]$	-49.72	-135.25	$-\infty$	-38.54
$[2,3]$	-17.78	-27.22	-42.68	-9.51
$[2.4,2.6]$	-10.25	-10.40	-9.01	-8.36

2.3 LP-Forms for Products

LP-forms can also be applied to products, although not directly. Consider the product $f(x) = \prod_j f_j(x)$, with all $f_j(x) \in C^1$ where, as for sums, we are given the inclusions $F_j(X) \supseteq f_j(X)$ and $F_j'(X) \supseteq f_j'(X)$ and the values $f_j(\underline{X})$ and $f_j(\overline{X})$ for all j. Following the reasoning of Lemma 3, the optimal inclusion function for $f(x)$ is

$$\begin{aligned} F^\diamond(x) &= \textstyle\prod_j F_j^\diamond(x) \\ &= \textstyle\prod_j \left[F_j(X) \bigcap \left(f_j(\underline{X}) + F_j'(X)(x - \underline{X}) \right) \bigcap \left(f_j(\overline{X}) + F_j'(X)(x - \overline{X}) \right) \right]. \end{aligned}$$

The difficulty with using $F^\diamond(x)$ is that the segmented lower and upper bounding functions generated by the products of $F_j^\diamond(x)$ are not linear and the envelope is not convex, in general, so in this case $F^\diamond(x)$ is not an LP-form. This precludes solving for $F^\diamond(X)$ by linear programming and leaves us with yet another difficult global optimization problem. We avoid this difficulty by finding an optimal inclusion for a transformed version of $f(x)$ that is amenable to linear programming, although in general this inclusion will be worse than $F^\diamond(X)$.

Assume, for now, that the $f_j(x)$ are known to be nonnegative on X, that is, $\underline{F}_j(X) \geq 0$. Nonpositive factors (and their inclusions) can be scaled by -1 and included, possibly changing the product being considered below to $-f(x)$. We apply a logarithmic transformation to $f(x)$, which gives

$$l(x) \doteq \ln(f(x)) = \sum_j \ln(f_j(x)) \doteq \sum_j l_j(x).$$

We can treat $l(x)$ exactly as the sums of the previous section. The optimal inclusion function for $l(x)$ on X is the LP-form

$$\begin{aligned} L^\diamond(x) &= \textstyle\sum_j L_j^\diamond(x) \\ &= \textstyle\sum_j \left[L_j(X) \bigcap \left(l_j(\underline{X}) + L_j'(X)(x - \underline{X}) \right) \bigcap \left(l_j(\overline{X}) + L_j'(X)(x - \overline{X}) \right) \right]. \end{aligned}$$

Note that $L^\diamond(x) \neq \ln(F^\diamond(x))$ in general.

By analogy to (2.2) we consider the representation

$$f(x) \;=\; \left(e^{\delta x}\prod_j f_j^{\alpha_j}(x)\right)\left(e^{-\delta x}\prod_j f_j^{\beta_j}(x)\right)\left(\prod_j f_j^{\gamma_j}(x)\right)$$

which transforms to

$$l(x) \;=\; \left(\sum_j \alpha_j l_j(x)+\delta x\right) \;+\; \left(\sum_j \beta_j l_j(x)-\delta x\right) \;+\; \left(\sum_j \gamma_j l_j(x)\right)$$

where $\alpha_j, \beta_j, \gamma_j, \delta \in \mathbf{R}$ and $\alpha_j + \beta_j + \gamma_j = 1$ for all j. The linear programs and their optimality theorems are identical to those presented for sums in Section 2.2 with l and L replacing f and F, respectively. The inclusion for $l(x)$ is turned into an inclusion for $f(x)$ by taking $F(X) = \exp(L^{\diamond}(X))$.

The values needed for these linear programs are easily obtained from the inclusions for the $f_j(x)$. Specifically,

$$l_j(\underline{X}) = \ln\Big(f_j(\underline{X})\Big),\;\; l_j(\overline{X}) = \ln\big(f_j(\overline{X})\big),\;\; L_j(X) = \ln\Big(F_j(X)\Big),\;\; \text{and}\;\; L_j'(X) = \tfrac{F_j'(X)}{F_j(X)}$$

where we extend $\ln(\cdot)$ to provide an exact inclusion for an interval argument. If $\underline{F}_j(X) = 0$, the range of $L_j'(X)$ may include $\pm\infty$ and we can set the corresponding α_j and/or β_j in the dual to zero. The interval division appearing in the inclusion for $l'(X)$ is the reason that the inclusion function $L^{\diamond}(x)$ is not generally as good as $F^{\diamond}(x)$. In some cases $l_j'(x) = f_j'(x)/f_j(x)$ may have a representation that gives a tighter inclusion than the one given above, and $L^{\diamond}(x)$ could be better than $F^{\diamond}(x)$.

Mixed sign factors can be treated with these methods by partitioning them into a sum of fixed sign factors. For example, if $f_m(X) \subseteq F_m(X) \subseteq [a, b]$ where $a < 0$ and $b > 0$, the product $p(x)f_m(x)$ becomes

$$p(x)\Big(f_m(x) - a\Big) + p(x)a \qquad \text{or} \qquad p(x)\Big(f_m(x) - b\Big) + p(x)b$$

where now each term is a product of fixed sign factors if $p(x)$ has fixed sign. Inclusions obtained from these sums of products may not be better than the product of the factor inclusions $P(X)F_m(X)$, particularly if $f_m(x)$ is not known monotonic on X or if $L_m'(X)$ includes $\pm\infty$.

The inclusion for $f(x)$ can be used to generate a good inclusion for $f'(x)$ by using the representation

$$f'(x) = f(x)\left(\sum_j l_j'(x)\right)$$

which gives the inclusion

$$F'(X) = F(X)\left(\sum_j L_j'(X)\right).$$

If the $L_j'(X)$ are bounded this may be a better inclusion than the "product rule" inclusion

$$F'(X) = \sum_i \left(F_i'(X) \prod_{j \neq i} F_j(X)\right).$$

2.4 Other Functions

The other nonunary factorable function operations we consider are exponentiation and taking the maximum or minimum of subfunctions. We can reduce exponentiation to the product case, for example

$$\Big(y(x)\Big)^{r(x)} \qquad \text{becomes} \qquad \exp\Big[r(x)\,\ln\Big(y(x)\Big)\Big]$$

which involves only unary functions and a product.

For the function $f(x) = \max_j (f_j(x))$, the maximum of convex functions is convex and the optimal lower bound $\underline{F}^\diamond(X)$ can be found by solving a linear program of the form

$$\min \;\; e$$

subject to

$$\begin{aligned}
e &\geq f_j(\underline{X}) + \underline{F}_j'(X)(x - \underline{X}) && \forall\, j \\
e &\geq f_j(\overline{X}) + \overline{F}_j'(X)(x - \overline{X}) && \forall\, j \\
e &\geq \underline{F}_j(X) && \forall\, j \\
x &\geq \underline{X} \\
x &\leq \overline{X}
\end{aligned}$$

The upper envelope of $f(x)$ is not concave but the upper bound $\overline{F}(X) = \max_j\big(\overline{F}_j(X)\big)$ is clearly optimal. The derivative inclusion is $F'(X) = \bigcup_j F_j'(X)$. Note that $f(x)$ is not, in general, a C^1 function (although it is continuous), so we are really using an inclusion of the subgradient of $f(x)$. This is discussed further in Section 5. The analogous results hold for taking a minimum of subfunctions.

Lastly, we consider the unary functions. Suppose $f(x) = u(y(x))$ and we have the inclusion (from an LP-form or otherwise) $Y(X) \supseteq y(X)$. The inclusion for $f(x)$ on X would then be simply $F(X) \doteq U(Y(X)) \supseteq u(Y(X))$ where $U(\cdot)$ is our inclusion for $u(\cdot)$.

No LP-forms are involved here; this is same approach used in forming the natural interval extension. Derivative inclusions $F'(X)$ can be obtained from the chain rule

$$f'(x) = \frac{du}{dy}\, y'(x).$$

3 Multivariate LP-Forms

LP-form inclusion functions extend in a natural way to multivariate functions. We consider each subfunction $f(\boldsymbol{x})$ on $\boldsymbol{X}$, where

$$\boldsymbol{x} = (x_1, \ldots, x_n) \qquad \text{and} \qquad \boldsymbol{X} = (X_1, \ldots, X_n).$$

For now we assume that, for propagating $f(\boldsymbol{x})$ to higher-level functions, we will simplify the envelope to use only a combination of values of $f(\boldsymbol{x})$, an inclusion for $f(\boldsymbol{X})$, and an inclusion for its gradient. We generalize the univariate approach by allowing an arbitrary set of values of $f(\boldsymbol{v})$, where $\boldsymbol{v}$ is a vertex of the box $\boldsymbol{X}$, and the vertex set is $\mathcal{V}(\boldsymbol{X}) = \big\{\boldsymbol{v} = (v_1, \ldots, v_n) : \, v_i \in \{\underline{X}_i, \overline{X}_i\}\big\}$. Our focus here will be to take envelopes of this form for the subfunctions of $f(\boldsymbol{x})$ and find an inclusion of $f(\boldsymbol{X})$.

Consider the function $f(\boldsymbol{x}) = \sum_j f_j(\boldsymbol{x})$ where we are given the information

$$\left(f_j\big(\boldsymbol{v}^{(k)}\big) \;\; \forall\, \boldsymbol{v}^{(k)} \in V \subseteq \mathcal{V}(\boldsymbol{X}), \quad \boldsymbol{G}_j(\boldsymbol{X}) \supseteq \boldsymbol{\nabla} f_j(\boldsymbol{X}), \quad \text{and} \;\; F_j(\boldsymbol{X}) \supseteq f_j(\boldsymbol{X}) \right) \quad \forall\, j \tag{3.1}$$

with which to compute the inclusion for $f(\boldsymbol{x})$. The gradient and its inclusion are taken to be row vectors here. By analogy to Lemma 3 for univariate functions, we can show that the optimal inclusion function for $f(\boldsymbol{x})$ on $\boldsymbol{X}$ is the LP-form

$$F^{\diamond}(\boldsymbol{x}) = \sum_j F_j^{\diamond}(\boldsymbol{x}) = \sum_j \; F_j(\boldsymbol{X}) \bigcap \left[\cap_k \left(f_j\big(\boldsymbol{v}^{(k)}\big) + \boldsymbol{G}_j(\boldsymbol{X}) \left(\boldsymbol{x} - \boldsymbol{v}^{(k)}\right) \right) \right].$$

The primal linear program for the lower bound is

$$\min \; \sum_j e_j$$

subject to

$$\begin{array}{rcll}
e_j & \geq & f_j\big(\boldsymbol{v}^{(k)}\big) + \underline{\boldsymbol{G}}_j^{(k)}(\boldsymbol{X}) \left(\boldsymbol{x} - \boldsymbol{v}^{(k)}\right) & \quad \forall\, k, \;\; \forall\, j \\
e_j & \geq & \underline{F}_j(\boldsymbol{X}) & \quad \forall\, j \\
x_i & \geq & \underline{X}_i & \quad \forall\, i \\
x_i & \leq & \overline{X}_i & \quad \forall\, i
\end{array}$$

where we define

$$\left(\underline{\boldsymbol{G}}_j^{(k)}(\boldsymbol{X})\right)_i \doteq \begin{cases} \underline{G}_{ji}(\boldsymbol{X}) \supseteq \underline{\nabla}_{ji}(\boldsymbol{X}) & \text{if} \quad v_i^{(k)} = \underline{X}_i \\ \overline{G}_{ji}(\boldsymbol{X}) \supseteq \overline{\nabla}_{ji}(\boldsymbol{X}) & \text{if} \quad v_i^{(k)} = \overline{X}_i. \end{cases}$$

The dual, which we need below, is

$$\max \quad \sum_k \left(\sum_j \alpha_j^{(k)} f_j\Big(\boldsymbol{v}^{(k)}\Big) \right) + \sum_j \gamma_j \underline{F}_j(\boldsymbol{X}) + \sum_i \delta_i \big(\underline{X}_i - \overline{X}_i\big) \tag{3.2}$$

subject to

$$\begin{array}{rcll} \sum_{k \in \underline{K}_i} \sum_j \alpha_j^{(k)} \, \underline{G}_{ji}^{(k)}(\boldsymbol{X}) + \delta_i & \geq & 0 & \forall\, i \\ \sum_{k \in \overline{K}_i} \sum_j \alpha_j^{(k)} \, \overline{G}_{ji}^{(k)}(\boldsymbol{X}) - \delta_i & \leq & 0 & \forall\, i \\ \sum_k \alpha_j^{(k)} + \gamma_j & = & 1 & \forall\, j \\ \alpha_j^{(k)},\ \gamma_j & \geq & 0 & \forall\, k,\ \forall\, j \end{array} \tag{3.3}$$

where

$$\underline{K}_i \doteq \Big\{k :\, \boldsymbol{v}^{(k)} \in V,\ v_i^{(k)} = \underline{X}_i\Big\} \qquad \text{and} \qquad \overline{K}_i \doteq \Big\{k :\, \boldsymbol{v}^{(k)} \in V,\ v_i^{(k)} = \overline{X}_i\Big\}.$$

The first two sets of constraints on the dual enforce the monotonicity of all objective terms that contain $\underline{X}_i$ or $\overline{X}_i$, respectively.

The multivariate LP-forms for products and the other types of functions are analogous generalizations of the univariate cases. We now consider improvements to this basic multivariate approach.

3.1 Generalizations

We can generalize the LP-form approach for multivariate functions in a number of ways. For one, there is no reason that we must use the same vertex set V for each subfunction in the LP-forms. There may be different advantageous choices V_j for each $f_j(\boldsymbol{x})$. We can also use different vertex sets for the upper and lower LP-form envelopes (the definition of $F^\diamond(\boldsymbol{x})$ can be modified to reflect this use of partial information).

A valuable modification of the LP-forms given above is to use a gradient inclusion that has fewer interval arguments than the full gradient inclusion $\boldsymbol{\nabla} f(\boldsymbol{X})$. This idea has been used in the algorithms of Hansen [5, 6]. We replace the gradient inclusion of (3.1) with an inclusion having components

$$G_i^{(k)}(\boldsymbol{X}) \supseteq \nabla_i f\Big(X_1, \ldots, X_i, v_{i+1}^{(k)}, \ldots, v_n^{(k)}\Big). \tag{3.4}$$

The inclusion function $f(\boldsymbol{v}^{(k)}) + \boldsymbol{G}(\boldsymbol{X})(\boldsymbol{x} - \boldsymbol{v}^{(k)})$ then becomes an inclusion of $f(\boldsymbol{x})$ on successively higher dimension subboxes of $\boldsymbol{X}$, that is,

$$f\Big(x_1, \ldots, x_m, v_{m+1}^{(k)}, \ldots, v_n^{(k)}\Big) \subseteq f\Big(\boldsymbol{v}^{(k)}\Big) + \sum_{i \leq m} \nabla_i f\Big(X_1, \ldots, X_i, v_{i+1}^{(k)}, \ldots, v_n^{(k)}\Big) \Big(x_i - v_i^{(k)}\Big)$$

for any $(x_1, \ldots, x_m) \in (X_1, \ldots, X_m)$.

3.2 Exploiting Monotonicity

The improved gradient inclusion of (3.4) can be used in an adaptive algorithm that produces better inclusions that are exact under a weak monotonicity condition. This property is based on reordering the variables x_i in the successive gradient expansion of (3.4).

Definition. A real-valued function $f(\boldsymbol{x}) \in C^1$ has the *lower successive monotonicity* property on $\boldsymbol{X}$ if there exists an ordering $x_1, \ldots, x_n$ such that $\nabla_i f\Big(X_1, \ldots, X_i, v_{i+1}^{(*)}, \ldots, v_n^{(*)}\Big)$ is either nonnegative or nonpositive for each i, where

$$v_i^{(*)} \doteq \begin{cases} \underline{X}_i & \text{if} \quad \nabla_i f\Big(X_1, \ldots, X_i, v_{i+1}^{(*)}, \ldots, v_n^{(*)}\Big) \geq 0 \\ \overline{X}_i & \text{if} \quad \nabla_i f\Big(X_1, \ldots, X_i, v_{i+1}^{(*)}, \ldots, v_n^{(*)}\Big) \leq 0. \end{cases} \tag{3.5}$$

The *upper successive monotonicity* property is defined analogously, with

$$v_i^{(*)} \doteq \begin{cases} \underline{X}_i & \text{if} \quad \nabla_i f\Big(X_1, \ldots, X_i, v_{i+1}^{(*)}, \ldots, v_n^{(*)}\Big) \leq 0 \\ \overline{X}_i & \text{if} \quad \nabla_i f\Big(X_1, \ldots, X_i, v_{i+1}^{(*)}, \ldots, v_n^{(*)}\Big) \geq 0. \end{cases} \tag{3.6}$$

Note that this is a weaker condition than monotonicity on $\boldsymbol{X}$, which requires $\nabla_i f(\boldsymbol{X})$ to be either nonnegative or nonpositive for each i. These definitions lead to an easy lemma showing that the $f(\boldsymbol{v}^{(*)})$ values are exact bounds on $f(\boldsymbol{X})$.

Lemma 5 *If $f(\boldsymbol{x}) \in C^1$ has the lower successive monotonicity property on $\boldsymbol{X}$, then $\underline{f}(\boldsymbol{X}) = f(\boldsymbol{v}^{(*)})$ with $\boldsymbol{v}^{(*)}$ defined in (3.5). If $f(\boldsymbol{x}) \in C^1$ has the upper successive monotonicity property on $\boldsymbol{X}$, then $\overline{f}(\boldsymbol{X}) = f(\boldsymbol{v}^{(*)})$ with $\boldsymbol{v}^{(*)}$ defined in (3.6).*

In practice we can readily test for these properties by first examining $G_i(\boldsymbol{X}) \supseteq \nabla_i f(\boldsymbol{X})$ for each i, selecting the appropriate coordinates of $\boldsymbol{v}^{(*)}$ for each i where $G_i(\boldsymbol{X})$ is nonnegative or nonpositive, and placing those coordinates at the end of the reordered list. Then we repeat this process for the subbox of $\boldsymbol{X}$ with the fixed coordinates until all coordinates of $\boldsymbol{v}$ have been fixed or no more monotonic directions remain. Let the set of all vertices matching the fixed coordinates be $V^{(*)}$ and the subbox of $\boldsymbol{X}$ matching the fixed coordinates be $\boldsymbol{X}^{(*)}$. As long as $G_i(\cdot)$ is inclusion monotonic, that is, if

$$\boldsymbol{X} \subseteq \boldsymbol{Y} \quad \Rightarrow \quad G_i(\boldsymbol{X}) \subseteq G_i(\boldsymbol{Y}),$$

this procedure will detect lower or upper successive monotonicity if it can be detected using the inclusions $G_i(\cdot)$. If the $G_i(\cdot)$ are Lipschitz inclusions, performing these monotonicity tests assures that an exact inclusion for $f(\cdot)$ will be obtained on a finite box containing each point $\hat{\boldsymbol{x}}$ where $\boldsymbol{\nabla} f(\hat{\boldsymbol{x}})$ has no zero components. For most C^1 functions this guarantees convergence to exact bounds in a finite number of steps everywhere but on a set of measure zero.

If successive monotonicity is detected and the appropriate $\boldsymbol{v}^{(*)}$ is in V, the LP-form bounds are also the exact value $f(\boldsymbol{v}^{(*)})$, as we would hope. For example, consider the dual lower bound linear program for a sum given in (3.2) and (3.3). Suppose that $f(\boldsymbol{x})$ is lower successive monotonic on $\boldsymbol{X}$ for the ordering $x_1, \ldots, x_n$ with the minimum occurring at $\boldsymbol{v}^{(*)}$, and that this has been detected, that is,

$$G_i^{(*)}\Big(X_1, \ldots, X_i, v_{i+1}^{(*)}, \ldots, v_n^{(*)}\Big) = \sum_j G_{ji}^{(*)}\Big(X_1, \ldots, X_i, v_{i+1}^{(*)}, \ldots, v_n^{(*)}\Big)$$

is either nonnegative or nonpositive for each i. It is not hard to verify that the following is a feasible solution to the dual

$$\Big(\alpha_j^{(*)} = 1,\ \alpha_j^{(k)} = 0 \ \forall\, \alpha_j^{(k)} \neq \alpha_j^{(*)}, \ \text{ and } \ \gamma_j = 0\Big) \quad \forall\, j, \qquad \text{and} \quad \delta_i = 0 \quad \forall\, i$$

and thus the corresponding objective value of $f(\boldsymbol{v}^{(*)})$ is a lower bound on $f(\boldsymbol{X})$. But since this value is actually achieved at $\boldsymbol{v}^{(*)} \in \boldsymbol{X}$ we can conclude that this is also the optimal value of the dual and the exact lower bound $\underline{f}(\boldsymbol{X})$.

If we fail to prove a successive monotonicity property, the fixed coordinates we obtain are still good in the sense that we restrict the lower or upper bounding problem to a subbox $\boldsymbol{X}^{(*)}$ where the desired extremum occurs. This reduces the effective dimension for all subsequent subboxes of $\boldsymbol{X}$. If the function is the objective function in our overall optimization problem then we can actually replace $\boldsymbol{X}$ by $\boldsymbol{X}^{(*)}$. This improves the quality of the bound as long as the overall bound process is itself inclusion monotonic. The linear programs can be simplified by using only vertices from the set $V^{(*)}$ and eliminating the variables x_i of the primal and δ_i of the dual corresponding to the fixed coordinates. If we are willing to pay the price of recursion, we can also use the better inclusions $F_j(\boldsymbol{X}^{(*)})$ instead of $F_j(\boldsymbol{X})$.

The validity of the simplification is demonstrated by showing that the optimal value of the dual can only increase when a vertex not in $V^{(*)}$ is replaced by the corresponding vertex of $V^{(*)}$, and then that the optimal value of δ_i for the fixed coordinates is zero. Alternately, it can be shown that the LP-form bounds are indeed inclusion monotonic.

4 General Polyhedral Envelopes

We can expect a better inclusion for $z(\boldsymbol{x})$ if we propagate the full LP-form envelopes through the function graph instead of simplifying the envelopes for each subfunction. This approaches a polyhedral analogue of the methods of McCormick [10], but without the difficulties arising from the need to maintain differentiability. The convex envelopes of unary functions used by McCormick are tighter than polyhedral envelopes, but our treatment of products is generally tighter. We will also use vertex gradient expansions at each stage to improve the envelopes and allow monotonicity to be exploited, as before.

Assume now that each function will be represented by an LP-form $F^\diamond(\boldsymbol{x}) \supseteq f(\boldsymbol{x})$ on a box $\boldsymbol{X}$. Given such an LP-form for each subfunction in $f(\boldsymbol{x}) = \sum_j f_j(\boldsymbol{x})$, the optimal LP-form for $f(\boldsymbol{x})$ is simply $F^\diamond(\boldsymbol{x}) = \sum_j F_j^\diamond(\boldsymbol{x})$. The primal linear programs for $f(\boldsymbol{X})$ contain all of the lower or upper constraints defining the $F_j^\diamond(\boldsymbol{x})$. If each $F_j^\diamond(\boldsymbol{x})$ has gradient expansion constraints for the same vertex set V, then gradient expansions from V are already included for $F^\diamond(\boldsymbol{x})$. A gradient inclusion for $f(\boldsymbol{x})$ better than $\sum_j \boldsymbol{G}_j(\boldsymbol{X})$ may be available. We may also want to use a different V for $f(\boldsymbol{x})$ than was used for the $f_j(\boldsymbol{x})$, particularly to exploit the monotonicities of $f(\boldsymbol{x})$. The simplest approach is just to allow any additional constraints to be added to $F^\diamond(\boldsymbol{x})$.

For the case of products we must handle the logarithmic transformation of the LP-forms $F_j^\diamond(\boldsymbol{x})$, which leaves a sum that we handle as above. The logarithm transform will be handled as an instance of a unary function. Before addressing the unary functions, we discuss the maximum and minimum of subfunctions. The lower envelope of a maximum and the upper envelope of a minimum are handled simply by combining the corresponding constraints, as in the univariate case of Section 2.4. The other halves of the envelopes must be convexified, and there is no simple method for doing this. The best approach is probably to use the simplified envelope analogous to the one given for the univariate case.

Finally, we consider unary functions. Suppose $f(\boldsymbol{x}) = u(y(\boldsymbol{x}))$ and we have the LP-form $Y^\diamond(\boldsymbol{x})$ for $y(\boldsymbol{x})$ on $\boldsymbol{X}$. We assume that the inclusion $Y^\diamond(\boldsymbol{X})$ has been computed and that we can compute a polyhedral envelope $U^\diamond(y) \supseteq u(y)$ on $Y^\diamond(\boldsymbol{X})$. Then an LP-form for $f(\boldsymbol{x})$ is $F^\diamond(\boldsymbol{x}) \doteq U^\diamond(Y^\diamond(\boldsymbol{x}))$ where we extend $U^\diamond(\cdot)$ to an inclusion in the natural fashion. It is not hard to show that a composition of LP-forms is always an LP-form.

In practice we would simply combine the constraints on $u(y)$ in $U^\diamond(y)$ with the constraints on $y(\boldsymbol{x})$ in $Y^\diamond(\boldsymbol{x})$, keeping the additional variable y. This effectively defines an LP-form $F^\diamond(y, \boldsymbol{x}) \doteq U^\diamond(y \cap Y^\diamond(\boldsymbol{x}))$ for which $F^\diamond(\boldsymbol{x}) = F^\diamond(Y(\boldsymbol{X}), \boldsymbol{x})$. In general this $F^\diamond(\boldsymbol{x})$ does not include gradient expansions from the values $u(y(\boldsymbol{v}))$ at vertices $\boldsymbol{v} \in \mathcal{V}(\boldsymbol{X})$, but these can be added to $F^\diamond(\boldsymbol{x})$.

This process introduces additional variables and the number of constraints grows at each level of the function graph, but the growth is only additive in the number of subfunctions. The improvement in the overall inclusion can easily justify the additional effort required to propagate these more accurate LP-forms.

5 Discussion

The LP-form approach to global optimization is really a class of methods for generating polyhedral envelope inclusions for real-valued functions. A variety of additions can be made to the envelope-defining constraints. We might, for example, wish to add expansions from facets of $\boldsymbol{X}$, such as $F(X_1, v_2)+G_2(\boldsymbol{X})(x_2-v_2)$ for a two-dimensional problem. The essence of the ideas presented is to use the available information to propagate tractable and accurate inclusion functions through the function graph.

There are a few ways to improve the gradient inclusions used in the LP-forms. First, for expansions from a particular vertex $\boldsymbol{v}$ we can replace an inclusion of the gradient with a slope inclusion

$$\hat{G}_i(\boldsymbol{X}) \supseteq \frac{f(\boldsymbol{x}) - f(x_1, \ldots, x_{i-1}, v_i, x_{i+1}, \ldots, x_n)}{x_i - v_i}$$

where we can restrict some components of $\boldsymbol{x}$ to match those of $\boldsymbol{v}$ to get a successive expansion that can be used in the same way as (3.4) with modified versions of the successive monotonicity properties. This requires a different set of slope inclusions for each vertex used, but the range of the slope is smaller than the corresponding gradient component and it may be possible to find good slope inclusions for some functions.

We can also apply the LP-form approach to finding good gradient inclusions. This requires Hessian bounds and increases the effort by an $\mathcal{O}(n)$ factor. The value of proving even partial successive monotonicity may make the effort justified. It may be possible to use the Hessian bounds directly in the LP-forms, but they are more suited to a nonpolyhedral method.

The results of Section 3.2 demonstrate that monotonicity is powerful and easily exploited within the LP-form inclusion process. The ability to obtain exact bounds in regions where the function is neither convex nor fully monotonic is surprising and welcome. More importantly, using partial monotonicities to reduce the effective dimension of the bounding problem may be crucial to the practical success of deterministic global optimization on medium- and high-dimension problems.

A number of extensions are possible. The methods presented can be applied to nondifferentiable functions if we replace the gradient inclusions with subgradient inclusions. The quadratic convergence is lost on boxes containing gradient discontinuities. We can allow discrete variables if we employ subgradients and perform a shrinking operation on each box generated.

Linear constraints on $\boldsymbol{x}$ can be directly incorporated into the LP-forms if we define $F(\emptyset) = \emptyset$ to indicate infeasible boxes. Linear constraints on the value of a subfunction are also easily included in the LP-form for the subfunction.

LP-forms can also be used for applications of interval analysis other than global optimization, such as solving nonlinear systems of equations.

References

[1] Alefeld, G. and Herzberger, J. *Introduction to Interval Computations.* Academic Press, New York (1983).

[2] Dixon, L.C.W. and Szego, G.P. eds. *Towards Global Optimisation.* North Holland, Amsterdam (1975).

[3] Dixon, L.C.W. and Szego, G.P. eds. *Towards Global Optimisation 2.* North Holland, Amsterdam (1978).

[4] Falk, J.E. and Soland, R.M. An algorithm for separable nonconvex programming problems. *Manag. Sci.* **15** 550-569 (1969).

[5] Hansen, E.R. On solving systems of equations using interval arithmetic. *Math. Comp.* **22** 374-384 (1968).

[6] Hansen, E.R. Global optimization using interval analysis - The multi-dimensional case. *Numer. Math.* **34** 247-270 (1980).

[7] Hansen, E.R. and Sengupta, S. Global constrained optimization using interval analysis. In *Interval Mathematics 1980*, ed. W. Nickel. Academic Press, New York (1980).

[8] Horst, R. An algorithm for nonconvex programming problems. *Math. Prog.* **10** 312-321 (1976).

[9] Kiefer, J. Sequential minimax search for a maximum. *Proc. Amer. Math. Soc.* **4** 502 (1953).

[10] McCormick, G.P. *Nonlinear Programming - Theory, Algorithms, and Applications.* Wiley, New York (1983).

[11] Moore, R.E. *Interval Analysis.* Prentice-Hall, Englewood Cliffs, New Jersey (1966).

[12] Moore, R.E. On computing the range of a rational function of n variables over a bounded region. *Computing* **16** 1-15 (1976).

[13] Moore, R.E. *Methods and Applications of Interval Analysis.* (SIAM Studies in Applied Mathematics). SIAM, Philadelphia (1979).

[14] Ratschek, H. and Rokne, J. *Computer Methods for the Range of Functions.* Ellis Horwood, Chichester (1984).

[15] Ratschek, H. and Rokne, J. *New Computer Methods for Global Optimization.* Ellis Horwood, Chichester (1988).

[16] Rinnooy Kan, A.H.G. and Timmer, G.T. *Global Optimization.* Report 8612/A, Econometric Institute, Erasmus University, Rotterdam (1986).

[17] Rosen, J.B. Global minimization of a linearly constrained concave function by partition of feasible domain. *Math. Oper. Res.* **8** 215-230 (1983).