Tools for Simplicial Branch and Bound in Global Optimization

Sam Karhbet
and
Ralph Baker Kearfott

Department of Mathematics
University of Louisiana at Lafayette

SCAN 2016, September 27, 2016
11:45–12:10

Outline

Introduction

Elements of Branch and Bound (B&B) algorithms

- Our goal is to minimize an objective function φ subject to various equality and inequality constraints, and to do this in a mathematically rigorous way.

Introduction

Elements of Branch and Bound (B&B) algorithms

- Our goal is to minimize an objective function φ subject to various equality and inequality constraints, and to do this in a mathematically rigorous way.
- The algorithm essentials relevant here are:

```
while Termination criteria are not met do
    Select a region 𝒟 from a list of unprocessed regions;
      Bound: Apply filters involving bounds on ranges to
      eliminate 𝒟 or portions of it from the search;
    if 𝒟 cannot be eliminated or stored then
        Branch: Split 𝒟 into two or more sub-regions
        whose union is 𝒟;
        Put each of the sub-regions into the list of
          unprocessed regions;
    end
end
```

Introduction

Bounding ranges of a function f over a region $\mathcal{D}$:

Is $\mathcal{D}$ a box or a simplex?

Introduction

Bounding ranges of a function f over a region $\mathcal{D}$:

Is $\mathcal{D}$ a box or a simplex?

- In most B&B algorithms, $\mathcal{D}$ is a box or *set of bounds on the coordinates*.

Introduction

Bounding ranges of a function f over a region $\mathcal{D}$:

Is $\mathcal{D}$ a box or a simplex?

- In most B&B algorithms, $\mathcal{D}$ is a box or *set of bounds on the coordinates*.
  - For boxes, bounds on the ranges of functions f can be computed rigorously with simple interval evaluations or with well-studied linear relaxations.

Introduction

Bounding ranges of a function f over a region $\mathcal{D}$:

Is $\mathcal{D}$ a box or a simplex?

- In most B&B algorithms, $\mathcal{D}$ is a box or *set of bounds on the coordinates*.
  - For boxes, bounds on the ranges of functions f can be computed rigorously with simple interval evaluations or with well-studied linear relaxations.
- In some problems, the natural region is an *n*-simplex *(e.g. a triangle for $n = 2$, a tetrahedron for $n = 3$, defined by $n + 1$ vertices)*, rather than a box.

Introduction

Bounding ranges of a function f over a region $\mathcal{D}$: Is $\mathcal{D}$ a box or a simplex?

- In most B&B algorithms, $\mathcal{D}$ is a box or *set of bounds on the coordinates*.
  - For boxes, bounds on the ranges of functions f can be computed rigorously with simple interval evaluations or with well-studied linear relaxations.
- In some problems, the natural region is an *n*-simplex *(e.g. a triangle for* $n = 2$*, a tetrahedron for* $n = 3$*, defined by* $n + 1$ *vertices)*, rather than a box.
  - Rigorously bounding ranges over a simplex has been less studied.

Introduction

Bounding ranges of a function f over a region $\mathcal{D}$:

Is $\mathcal{D}$ a box or a simplex?

- In most B&B algorithms, $\mathcal{D}$ is a box or *set of bounds on the coordinates*.
  - For boxes, bounds on the ranges of functions f can be computed rigorously with simple interval evaluations or with well-studied linear relaxations.
- In some problems, the natural region is an *n*-simplex *(e.g. a triangle for $n = 2$, a tetrahedron for $n = 3$, defined by $n + 1$ vertices)*, rather than a box.
  - Rigorously bounding ranges over a simplex has been less studied.
  - Two different representations of a simplex are useful in B&B algorithms, and how do we convert between these representations?

Related Work

Simplicial B&B and range computation over simplices

- Stenger, Kearfott, Stynes (1970's)

Related Work

Simplicial B&B and range computation over simplices

- Stenger, Kearfott, Stynes (1970's)
  - These works used a B&B algorithm based on simplicial subdivision to compute the topological degree of maps.

Related Work

Simplicial B&B and range computation over simplices

- Stenger, Kearfott, Stynes (1970’s)
  - These works used a B&B algorithm based on simplicial subdivision to compute the topological degree of maps.
  - These used heuristics to bound ranges.

Related Work

Simplicial B&B and range computation over simplices

- Stenger, Kearfott, Stynes (1970's)
  - These works used a B&B algorithm based on simplicial subdivision to compute the topological degree of maps.
  - These used heuristics to bound ranges.
- Garloff and others (1980's to the present)

Related Work

Simplicial B&B and range computation over simplices

- Stenger, Kearfott, Stynes (1970's)
  - These works used a B&B algorithm based on simplicial subdivision to compute the topological degree of maps.
  - These used heuristics to bound ranges.
- Garloff and others (1980's to the present)
  - Mathematically rigorous range computation of polynomials and rational functions is done with Bernstein polynomial expansions over simplices.

Related Work

Simplicial B&B and range computation over simplices

- Stenger, Kearfott, Stynes (1970's)
  - These works used a B&B algorithm based on simplicial subdivision to compute the topological degree of maps.
  - These used heuristics to bound ranges.
- Garloff and others (1980's to the present)
  - Mathematically rigorous range computation of polynomials and rational functions is done with Bernstein polynomial expansions over simplices.
  - Nataraj has used this approach to range computation of polynomials, but, to our knowledge, not over simplices.

Related Work

Simplicial B&B and range computation over simplices

- Stenger, Kearfott, Stynes (1970's)
  - These works used a B&B algorithm based on simplicial subdivision to compute the topological degree of maps.
  - These used heuristics to bound ranges.
- Garloff and others (1980's to the present)
  - Mathematically rigorous range computation of polynomials and rational functions is done with Bernstein polynomial expansions over simplices.
  - Nataraj has used this approach to range computation of polynomials, but, to our knowledge, not over simplices.
  - We are looking forward to investigation of the relative efficiency efficiency of these techniques.
- Paulavičius, Žilinskas, et al (current)

Related Work

Simplicial B&B and range computation over simplices

- Stenger, Kearfott, Stynes (1970's)
  - These works used a B&B algorithm based on simplicial subdivision to compute the topological degree of maps.
  - These used heuristics to bound ranges.
- Garloff and others (1980's to the present)
  - Mathematically rigorous range computation of polynomials and rational functions is done with Bernstein polynomial expansions over simplices.
  - Nataraj has used this approach to range computation of polynomials, but, to our knowledge, not over simplices.
  - We are looking forward to investigation of the relative efficiency efficiency of these techniques.
- Paulavičius, Žilinskas, et al (current)
  - They have extensively studied use of simplices in B&B algorithms for optimization.

Related Work

Simplicial B&B and range computation over simplices

- Stenger, Kearfott, Stynes (1970's)
  - These works used a B&B algorithm based on simplicial subdivision to compute the topological degree of maps.
  - These used heuristics to bound ranges.
- Garloff and others (1980's to the present)
  - Mathematically rigorous range computation of polynomials and rational functions is done with Bernstein polynomial expansions over simplices.
  - Nataraj has used this approach to range computation of polynomials, but, to our knowledge, not over simplices.
  - We are looking forward to investigation of the relative efficiency efficiency of these techniques.
- Paulavičius, Žilinskas, et al (current)
  - They have extensively studied use of simplices in B&B algorithms for optimization.
  - However, their published results involve heuristic or probabilistic bounds for ranges.

Two Simplex Representations

Vertex and halfspace representations

Two Simplex Representations

Vertex and halfspace representations

- The vertex representation of a simplex $\mathcal{D} = \mathcal{S}$ is in terms of the cartesian coordinates of its $n+1$ vertices, i.e. $\mathcal{S} = \langle P_0, P_1, \ldots P_n \rangle$.

Two Simplex Representations

Vertex and halfspace representations

- The vertex representation of a simplex $\mathcal{D} = \mathcal{S}$ is in terms of the cartesian coordinates of its $n + 1$ vertices, i.e. $\mathcal{S} = \langle P_0, P_1, \ldots P_n \rangle$.
- The half-plane representation of a simplex is in terms of the feasible set of $n + 1$ inequalities $Ax \geq b$, $A \in \mathbb{R}^{n+1 \times n}$, $b \in \mathbb{R}^{n+1}$.

Two Simplex Representations

Vertex and halfspace representations

- The vertex representation of a simplex $\mathcal{D} = \mathcal{S}$ is in terms of the cartesian coordinates of its $n+1$ vertices, i.e. $\mathcal{S} = \langle P_0, P_1, \ldots P_n \rangle$.
- The half-plane representation of a simplex is in terms of the feasible set of $n+1$ inequalities $Ax \geq b$, $A \in \mathbb{R}^{n+1 \times n}$, $b \in \mathbb{R}^{n+1}$.
  - Each *face* $\mathcal{S}_{\neg i}$ of $\mathcal{S}$ opposite a vertex P_i of $\mathcal{S}$ is contained in a hyperplane $\tilde{A}_{i,:} x = b_i$, where $\tilde{A}_{i,:} = \pm A_{i,:}$.

Two Simplex Representations

Vertex and halfspace representations

- The vertex representation of a simplex $\mathcal{D} = \mathcal{S}$ is in terms of the cartesian coordinates of its $n+1$ vertices, i.e. $\mathcal{S} = \langle P_0, P_1, \ldots P_n \rangle$.
- The half-plane representation of a simplex is in terms of the feasible set of $n+1$ inequalities $Ax \geq b$, $A \in \mathbb{R}^{n+1 \times n}$, $b \in \mathbb{R}^{n+1}$.
  - Each *face* $\mathcal{S}_{\neg i}$ of $\mathcal{S}$ opposite a vertex P_i of $\mathcal{S}$ is contained in a hyperplane $\tilde{A}_{i,:}x = b_i$, where $\tilde{A}_{i,:} = \pm A_{i,:}$.
  - The side of the hyperplane upon which P_i lies determines the sense of the inequality $A_{i,:}x \geq b_i$.

Two Simplex Representations

Vertex and halfspace representations

- The vertex representation of a simplex $\mathcal{D} = \mathcal{S}$ is in terms of the cartesian coordinates of its $n+1$ vertices, i.e. $\mathcal{S} = \langle P_0, P_1, \ldots P_n \rangle$.
- The half-plane representation of a simplex is in terms of the feasible set of $n+1$ inequalities $Ax \geq b$, $A \in \mathbb{R}^{n+1 \times n}$, $b \in \mathbb{R}^{n+1}$.
  - Each *face* $\mathcal{S}_{\neg i}$ of $\mathcal{S}$ opposite a vertex P_i of $\mathcal{S}$ is contained in a hyperplane $\tilde{A}_{i,:}x = b_i$, where $\tilde{A}_{i,:} = \pm A_{i,:}$.
  - The side of the hyperplane upon which P_i lies determines the sense of the inequality $A_{i,:}x \geq b_i$.
- The vertex representation is most useful in the branching, etc., while the halfspace representation is most useful in constraint-propagation-based filters.

Two Simplex Representations

Vertex and halfspace representations

- The vertex representation of a simplex $\mathcal{D} = \mathcal{S}$ is in terms of the cartesian coordinates of its $n+1$ vertices, i.e. $\mathcal{S} = \langle P_0, P_1, \ldots P_n \rangle$.
- The half-plane representation of a simplex is in terms of the feasible set of $n+1$ inequalities $Ax \geq b$, $A \in \mathbb{R}^{n+1 \times n}$, $b \in \mathbb{R}^{n+1}$.
  - Each *face* $\mathcal{S}_{\neg i}$ of $\mathcal{S}$ opposite a vertex P_i of $\mathcal{S}$ is contained in a hyperplane $\tilde{A}_{i,:}x = b_i$, where $\tilde{A}_{i,:} = \pm A_{i,:}$.
  - The side of the hyperplane upon which P_i lies determines the sense of the inequality $A_{i,:}x \geq b_i$.
- The vertex representation is most useful in the branching, etc., while the halfspace representation is most useful in constraint-propagation-based filters.
- We have studied mathematically rigorous conversions between these two representations.

Bounding the Range of f Over a Simplex

f is normally represented in terms of cartesian (box-based) coordinates.

Various possibilities

Bounding the Range of f Over a Simplex

f is normally represented in terms of cartesian (box-based) coordinates.

Various possibilities

- We can enclose $\mathcal{S}$ in a box, then use traditional interval extensions over the box.

Bounding the Range of f Over a Simplex

f is normally represented in terms of cartesian (box-based) coordinates.

Various possibilities

- We can enclose $\mathcal{S}$ in a box, then use traditional interval extensions over the box.
  - This is simple, but with significant overestimation.

Bounding the Range of f Over a Simplex

f is normally represented in terms of cartesian (box-based) coordinates.

Various possibilities

- We can enclose $\mathcal{S}$ in a box, then use traditional interval extensions over the box.
  - This is simple, but with significant overestimation.
- We can use the halfspace representation and constraint propagation.

Bounding the Range of f Over a Simplex

f is normally represented in terms of cartesian (box-based) coordinates.

Various possibilities

- We can enclose $\mathcal{S}$ in a box, then use traditional interval extensions over the box.
  - This is simple, but with significant overestimation.
- We can use the halfspace representation and constraint propagation.
  - This can result in less overestimation, but not necessarily.

Bounding the Range of f Over a Simplex

f is normally represented in terms of cartesian (box-based) coordinates.

Various possibilities

- We can enclose $\mathcal{S}$ in a box, then use traditional interval extensions over the box.
  - This is simple, but with significant overestimation.
- We can use the halfspace representation and constraint propagation.
  - This can result in less overestimation, but not necessarily.
  - This adds complication and, depending on the problem and how implemented, could involve an amount of computation comparable to that required to totally solve the original problem

Bounding the Range of f Over a Simplex

f is normally represented in terms of cartesian (box-based) coordinates.

Various possibilities

- We can enclose $\mathcal{S}$ in a box, then use traditional interval extensions over the box.
  - This is simple, but with significant overestimation.
- We can use the halfspace representation and constraint propagation.
  - This can result in less overestimation, but not necessarily.
  - This adds complication and, depending on the problem and how implemented, could involve an amount of computation comparable to that required to totally solve the original problem
- We can analyze relationships between coordinates in the simplex to derive simple formulas that give sharper bounds than interval extensions over the containing boxes.

Bounding the Range of f Over a Simplex $\mathcal{S}$

A specially derived formula for $\mathcal{S} = \langle P_0, P_1, \ldots P_n \rangle$, with $P_i = (p_{i,1}, \ldots, p_{i,n})$

Begin with non-sharp bounds $\boldsymbol{f} = [\underline{f}, \overline{f}]$, say, obtained by evaluating over a box $\boldsymbol{x}$ containing $\mathcal{S}$.

Bounding the Range of f Over a Simplex $\mathcal{S}$

A specially derived formula for $\mathcal{S} = \langle P_0, P_1, \ldots P_n \rangle$, with $P_i = (p_{i,1}, \ldots, p_{i,n})$

Begin with non-sharp bounds $\boldsymbol{f} = [\underline{f}, \overline{f}]$, say, obtained by evaluating over a box $\boldsymbol{x}$ containing $\mathcal{S}$.

Theorem:

$$\textit{Let } L_i = Inf\left(\sum_{j=1}^{n} p_{i,j} \check{f}_j(sgn(p_{i,j}))\right), \textit{ and}$$
$$U_i = Sup\left(\sum_{j=1}^{n} p_{i,j} \check{f}_j(-sgn(p_{i,j}))\right), \textit{ where}$$

$$\check{f}_j(p) = \left\{ \underline{f}_j \textit{ if } p \geq 0, \ \overline{f}_j \textit{ if } p < 0, \quad \textit{and} \quad \boldsymbol{f}_j \textit{ if } 0 \in p \right\}.$$

Bounding the Range of f Over a Simplex $\mathcal{S}$

A specially derived formula for $\mathcal{S} = \langle P_0, P_1, \ldots P_n \rangle$, with $P_i = (p_{i,1}, \ldots, p_{i,n})$

Begin with non-sharp bounds $\boldsymbol{f} = [\underline{f}, \overline{f}]$, say, obtained by evaluating over a box $\boldsymbol{x}$ containing $\mathcal{S}$.

Theorem:

Let
$$L_i = Inf\left(\sum_{j=1}^n p_{i,j}\check{f}_j(\ \ sgn(p_{i,j}))\right), \text{ and}$$
$$U_i = Sup\left(\sum_{j=1}^n p_{i,j}\check{f}_j(-sgn(p_{i,j}))\right), \text{ where}$$
$$\check{f}_j(p) = \left\{\underline{f}_j \text{ if } p \geq 0,\ \ \overline{f}_j \text{ if } p < 0, \quad \text{and} \quad \boldsymbol{f}_j \text{ if } 0 \in p\right\}.$$

Assume the domain of f has been translated so the barycenter $\frac{1}{n+1}\sum_{i=0}^n P_i$ is the origin $(0, \ldots, 0)$, and the range of f has been translated so $f(0, \ldots, 0) = 0$.

Bounding the Range of f Over a Simplex $\mathcal{S}$

A specially derived formula for $\mathcal{S} = \langle P_0, P_1, \ldots P_n \rangle$, with $P_i = (p_{i,1}, \ldots, p_{i,n})$

Begin with non-sharp bounds $\boldsymbol{f} = [\underline{f}, \overline{f}]$, say, obtained by evaluating over a box $\boldsymbol{x}$ containing $\mathcal{S}$.

Theorem:

Let

$$L_i = \mathit{Inf}\left(\sum_{j=1}^{n} p_{i,j} \check{f}_j(\mathit{sgn}(p_{i,j})) \right), \text{ and}$$

$$U_i = \mathit{Sup}\left(\sum_{j=1}^{n} p_{i,j} \check{f}_j(-\mathit{sgn}(p_{i,j})) \right), \text{ where}$$

$$\check{f}_j(p) = \left\{ \underline{f}_j \text{ if } p \geq 0, \ \overline{f}_j \text{ if } p < 0, \ \text{ and } \ \boldsymbol{f}_j \text{ if } 0 \in p \right\}.$$

Assume the domain of f has been translated so the barycenter $\frac{1}{n+1} \sum_{i=0}^{n} P_i$ is the origin $(0, \ldots, 0)$, and the range of f has been translated so $f(0, \ldots, 0) = 0$.

Then the range of f over $\mathcal{S}$ is contained in the interval $\boldsymbol{I}_0 = [\min_{0 \leq i \leq n} L_i, \max_{0 \leq i \leq n} U_i]$.

Bounding the Range of f Over a Simplex $\mathcal{S}$

A specially derived formula for $\mathcal{S} = \langle P_0, P_1, \ldots P_n \rangle$, with $P_i = (p_{i,1}, \ldots, p_{i,n})$

Begin with non-sharp bounds $\boldsymbol{f} = [\underline{f}, \overline{f}]$, say, obtained by evaluating over a box $\boldsymbol{x}$ containing $\mathcal{S}$.

Theorem:

Let
$$L_i = Inf\left(\sum_{j=1}^{n} p_{i,j}\check{f}_j(\;sgn(p_{i,j}))\right), \text{ and}$$
$$U_i = Sup\left(\sum_{j=1}^{n} p_{i,j}\check{f}_j(-sgn(p_{i,j}))\right), \text{ where}$$
$$\check{f}_j(p) = \left\{\underline{f}_j \text{ if } p \geq 0,\ \overline{f}_j \text{ if } p < 0, \text{ and } \boldsymbol{f}_j \text{ if } 0 \in p\right\}.$$

Assume the domain of f has been translated so the barycenter $\frac{1}{n+1}\sum_{i=0}^{n} P_i$ is the origin $(0, \ldots, 0)$, and the range of f has been translated so $f(0, \ldots, 0) = 0$.

Then the range of f over $\mathcal{S}$ is contained in the interval $\boldsymbol{I}_0 = [\min_{0\leq i\leq n} L_i, \max_{0\leq i \leq n} U_i]$.

- $\boldsymbol{I}_0$ is often narrower than $\boldsymbol{f}$.

Bounding the Range of f Over a Simplex $\mathcal{S}$

A specially derived formula for $\mathcal{S} = \langle P_0, P_1, \ldots P_n \rangle$, with $P_i = (p_{i,1}, \ldots, p_{i,n})$

Begin with non-sharp bounds $\boldsymbol{f} = [\underline{f}, \overline{f}]$, say, obtained by evaluating over a box $\boldsymbol{x}$ containing $\mathcal{S}$.

Theorem:

Let $L_i = Inf\left(\sum_{j=1}^{n} p_{i,j} \check{f}_j(\; sgn(p_{i,j})) \right)$, *and*

$U_i = Sup\left(\sum_{j=1}^{n} p_{i,j} \check{f}_j(-sgn(p_{i,j})) \right)$, *where*

$$\check{f}_j(p) = \left\{ \underline{f}_j \text{ if } p \geq 0, \; \overline{f}_j \text{ if } p < 0, \; \text{ and } \; \boldsymbol{f}_j \text{ if } 0 \in p \right\}.$$

Assume the domain of f has been translated so the barycenter $\frac{1}{n+1} \sum_{i=0}^{n} P_i$ is the origin $(0, \ldots, 0)$, and the range of f has been translated so $f(0, \ldots, 0) = 0$.

Then the range of f over $\mathcal{S}$ is contained in the interval $\boldsymbol{I}_0 = [\min_{0 \leq i \leq n} L_i, \max_{0 \leq i \leq n} U_i]$.

- $\boldsymbol{I}_0$ is often narrower than $\boldsymbol{f}$.

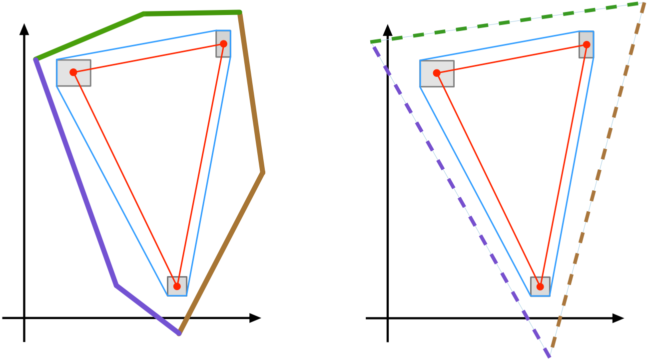
(The theorem is proven by considering $\mathcal{S}$ in terms of barycentric coordinates and an associated LP.)

The Vertex and Halfspace Representations

Computing a rigorous enclosure of $\mathcal{S}$ in a halfspace representation from a rigorous enclosure for $\mathcal{S}$ in a vertex representation

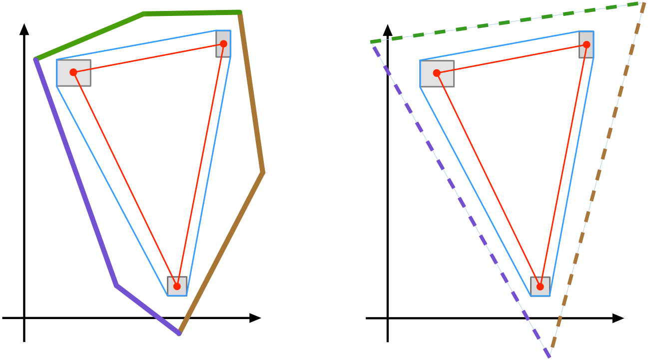
The Vertex and Halfspace Representations

Computing a rigorous enclosure of $\mathcal{S}$ in a halfspace representation from a rigorous enclosure for $\mathcal{S}$ in a vertex representation



The Vertex and Halfspace Representations

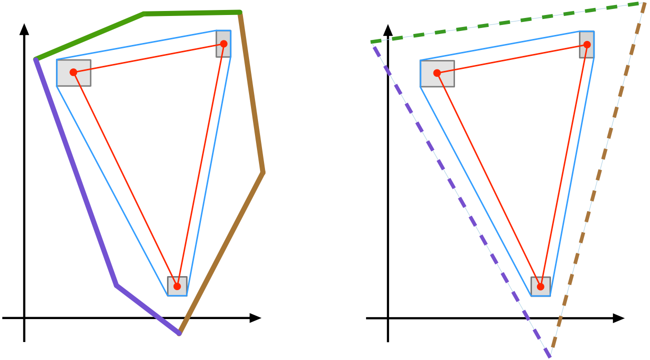
Computing a rigorous enclosure of $\mathcal{S}$ in a halfspace representation from a rigorous enclosure for $\mathcal{S}$ in a vertex representation



- We bound the set of all possible halfplane equations subject to uncertainties in the vertices.

The Vertex and Halfspace Representations

Computing a rigorous enclosure of $\mathcal{S}$ in a halfspace representation from a rigorous enclosure for $\mathcal{S}$ in a vertex representation



- We bound the set of all possible halfplane equations subject to uncertainties in the vertices.
- We select certain halfplanes arbitrarily to construct the system $Ax \geq b$.

Vertex Enclosure to Halfspace Enclosure

The computations for the i-th halfspace, $0 \le i \le n$
corresponding to $\mathcal{S}_{\neg i} = \left\langle \tilde{P}_0, \tilde{P}_1, \ldots, \tilde{P}_{n-1} \right\rangle$

Vertex Enclosure to Halfspace Enclosure

The computations for the i-th halfspace, $0 \le i \le n$ corresponding to $\mathcal{S}_{\neg i} = \left\langle \tilde{P}_0, \tilde{P}_1, \ldots, \tilde{P}_{n-1} \right\rangle$

- Begin with enclosures $\tilde{P}_i$ to the actual vertices $\tilde{P}_i$.

Vertex Enclosure to Halfspace Enclosure

The computations for the i-th halfspace, $0 \le i \le n$ corresponding to $\mathcal{S}_{\neg i} = \left\langle \tilde{P}_0, \tilde{P}_1, \ldots, \tilde{P}_{n-1} \right\rangle$

- Begin with enclosures $\tilde{\boldsymbol{P}}_i$ to the actual vertices $\tilde{P}_i$.
- For the i-th row of A, consider an interval enclosure to the system

$$\boldsymbol{M} a_i = \begin{pmatrix} (\tilde{\boldsymbol{P}}_1 - \tilde{\boldsymbol{P}}_0)^T \\ \vdots \\ (\tilde{\boldsymbol{P}}_{n-1} - \tilde{\boldsymbol{P}}_0)^T \end{pmatrix} a_i = 0.$$

Vertex Enclosure to Halfspace Enclosure

The computations for the i-th halfspace, $0 \le i \le n$ corresponding to $\mathcal{S}_{\neg i} = \left\langle \tilde{P}_0, \tilde{P}_1, \ldots, \tilde{P}_{n-1} \right\rangle$

- Begin with enclosures $\tilde{\boldsymbol{P}}_i$ to the actual vertices $\tilde{P}_i$.
- For the i-th row of A, consider an interval enclosure to the system

$$\boldsymbol{M}a_i = \begin{pmatrix} (\tilde{\boldsymbol{P}}_1 - \tilde{\boldsymbol{P}}_0)^T \\ \vdots \\ (\tilde{\boldsymbol{P}}_{n-1} - \tilde{\boldsymbol{P}}_0)^T \end{pmatrix} a_i = 0.$$

- We obtain a floating point approximation z to $M\check{a}_i = 0$, $\|z\|_2 = 1$ using a common null-space-finding procedure.

Vertex Enclosure to Halfspace Enclosure

The computations for the i-th halfspace, $0 \le i \le n$ corresponding to $\mathcal{S}_{\neg i} = \left\langle \tilde{P}_0, \tilde{P}_1, \ldots, \tilde{P}_{n-1} \right\rangle$

- Begin with enclosures $\tilde{\boldsymbol{P}}_i$ to the actual vertices $\tilde{P}_i$.
- For the i-th row of A, consider an interval enclosure to the system

$$\boldsymbol{M} a_i = \begin{pmatrix} (\tilde{\boldsymbol{P}}_1 - \tilde{\boldsymbol{P}}_0)^T \\ \vdots \\ (\tilde{\boldsymbol{P}}_{n-1} - \tilde{\boldsymbol{P}}_0)^T \end{pmatrix} a_i = 0.$$

- We obtain a floating point approximation z to $M\check{a}_i = 0$, $\|z\|_2 = 1$ using a common null-space-finding procedure.
- We construct a sufficiently large box $\boldsymbol{a}^{(0)}$ around z, and apply an interval Newton method to the system $\boldsymbol{M}z = 0$, $z^T z = 1$ to prove a unique solution for every $M \in \boldsymbol{M}$ and generating an enclosure $\boldsymbol{a}_i$ for the normal vector perpendicular to $\mathcal{S}_{\neg i}$.

Vertex Enclosure to Halfspace Enclosure

Computations for the *i*-th halfspace (continued)

- We possibly reverse the sign of $\boldsymbol{a}_i$ depending on the sign of $\boldsymbol{a}_i^T(\tilde{\boldsymbol{P}}_i - \boldsymbol{P}_0)$.

Vertex Enclosure to Halfspace Enclosure

Computations for the *i*-th halfspace (continued)

- We possibly reverse the sign of $\boldsymbol{a}_i$ depending on the sign of $\boldsymbol{a}_i^T(\tilde{\boldsymbol{P}}_i - \boldsymbol{P}_0)$.
- Compute $b_i \approx a_i^T \tilde{P}_0$ using floating point computations.

Vertex Enclosure to Halfspace Enclosure

Computations for the i-th halfspace (continued)

- We possibly reverse the sign of $\boldsymbol{a}_i$ depending on the sign of $\boldsymbol{a}_i^T(\tilde{\boldsymbol{P}}_i - \boldsymbol{P}_0)$.
- Compute $b_i \approx a_i^T \tilde{P}_0$ using floating point computations.
- Gradually decrease b_i until a $\underline{b_i}$ with $\boldsymbol{a}_i^T \boldsymbol{P}_j \geq \underline{b_i}$ for $0 \leq j \leq n$.

Vertex Enclosure to Halfspace Enclosure

Computations for the *i*-th halfspace (continued)

- We possibly reverse the sign of $\boldsymbol{a}_i$ depending on the sign of $\boldsymbol{a}_i^T(\tilde{\boldsymbol{P}}_i - \boldsymbol{P}_0)$.
- Compute $b_i \approx a_i^T \tilde{P}_0$ using floating point computations.
- Gradually decrease b_i until a $\underline{b}_i$ with $\boldsymbol{a}_i^T \boldsymbol{P}_j \geq \underline{b}_i$ for $0 \leq j \leq n$.
- Proposition: *Let* $\boldsymbol{H}_i = \{x : \boldsymbol{a}_i^T x \geq \underline{b}_i\}$. *Verification of* $\boldsymbol{a}_i^T \boldsymbol{P}_j \geq \underline{b}_i$ $(j = 0, 1, \ldots, n)$ *implies* $\mathcal{S} \subset \boldsymbol{H}_i$.

Vertex Enclosure to Halfspace Enclosure

Computations for the *i*-th halfspace (continued)

- We possibly reverse the sign of $\boldsymbol{a}_i$ depending on the sign of $\boldsymbol{a}_i^T(\tilde{\boldsymbol{P}}_i - \boldsymbol{P}_0)$.
- Compute $b_i \approx a_i^T \tilde{P}_0$ using floating point computations.
- Gradually decrease b_i until a $\underline{b}_i$ with $\boldsymbol{a}_i^T \boldsymbol{P}_j \geq \underline{b}_i$ for $0 \leq j \leq n$.
- Proposition: *Let* $\boldsymbol{H}_i = \{x : \boldsymbol{a}_i^T x \geq \underline{b}_i\}$*. Verification of* $\boldsymbol{a}_i^T \boldsymbol{P}_j \geq \underline{b}_i$ $(j = 0, 1, \ldots, n)$ *implies* $\mathcal{S} \subset \boldsymbol{H}_i$.
- Since $\boldsymbol{a}_i^T \boldsymbol{P}_j \geq \underline{b}_i$, $a_i^T \boldsymbol{P}_j \geq \underline{b}_i$ for *any* $a_i \in \boldsymbol{a}_i$, so, with the same reasoning behind the proposition, $\mathcal{S} \subset H_i = \{x : a_i^T x \geq \underline{b}_i\}$.

Vertex Enclosure to Halfspace Enclosure

Computations for the *i*-th halfspace (continued)

- We possibly reverse the sign of $\boldsymbol{a}_i$ depending on the sign of $\boldsymbol{a}_i^T(\tilde{\boldsymbol{P}}_i - \boldsymbol{P}_0)$.
- Compute $b_i \approx a_i^T \tilde{P}_0$ using floating point computations.
- Gradually decrease b_i until a $\underline{b}_i$ with $\boldsymbol{a}_i^T \boldsymbol{P}_j \geq \underline{b}_i$ for $0 \leq j \leq n$.
- Proposition: *Let* $\boldsymbol{H}_i = \{x : \boldsymbol{a}_i^T x \geq \underline{b}_i\}$*. Verification of* $\boldsymbol{a}_i^T \boldsymbol{P}_j \geq \underline{b}_i$ $(j = 0, 1, \ldots, n)$ *implies* $\mathcal{S} \subset \boldsymbol{H}_i$*.*
- Since $\boldsymbol{a}_i^T \boldsymbol{P}_j \geq \underline{b}_i$, $a_i^T \boldsymbol{P}_j \geq \underline{b}_i$ for *any* $a_i \in \boldsymbol{a}_i$, so, with the same reasoning behind the proposition, $\mathcal{S} \subset H_i = \{x : a_i^T x \geq \underline{b}_i\}$.
- In other words, a_i can be *any* floating-point quantity in $\boldsymbol{a}_i$.

What next?

Comparisons of simplicial-based and box-based B&B

- Sam has initial implementations of the same basic B&B algorithm using both simplices and boxes, incorporating the techniques we have explained here.

What next?

Comparisons of simplicial-based and box-based B&B

- Sam has initial implementations of the same basic B&B algorithm using both simplices and boxes, incorporating the techniques we have explained here.
- We have selected both general test problems and test problems on which there is an underlying simplicial geometry.

What next?

Comparisons of simplicial-based and box-based B&B

- Sam has initial implementations of the same basic B&B algorithm using both simplices and boxes, incorporating the techniques we have explained here.
- We have selected both general test problems and test problems on which there is an underlying simplicial geometry.
- This work is in progress.