The Linear Ranking Problem for Integer Linear-Constraint Loops

Amir Ben-Amram
The Academic College of Tel-Aviv Yaffo

and

Samir Genaim
Complutense University of Madrid

* Presented at POPL’13
Outline

- Background & problem presentation
- Decidability & complexity
- Tractable sub-problems
The problem is undecidable in general, but we can look for decidable sub-problems.

1) abstracting to simpler programs
2) restricting to certain termination proofs
The problem is undecidable in general, but we can look for decidable sub-problems.

1) abstracting to simpler programs
2) restricting to certain termination proofs
Abstraction of Programs for Termination Analysis

1. Define the abstract state space $S$. A typical state: a finite collection of integer variables $(x_1, ..., x_n)$

2. Choose a language for describing transitions in $S \times S$ (often over-approximating)
   - a class of formulas over $x_1, ..., x_n, x'_1, ..., x'_n$
The Linear-Constraint Abstraction

Variables are integers (could also be rationals / real)

Transitions are described by conjunctions of linear constraints (inequalities).

For such program (over integers), termination is still undecidable. They can represent counter-machines precisely.
Linear-Constraint Loops

One program location

\[ 1 \leq z \]

\[ x' = x + 1, \quad y' = y - 1, \quad z' = z - 1 \]

\[ \text{while} \ (z \geq 1) \ \text{do} \]
\[ x := x + 1; \]
\[ y := y - 1; \]
\[ z := z - 1; \]

One abstract transition - Single-path loop

Several transitions - Multiple-path loop
Linear-Constraint Loops

One program location

\[ 1 \leq z \]
\[ x' = x + 1, \ y' = y - 1, \ z' = z - 1 \]

while \((z \geq 1)\) do
\[ x := x + 1; \]
\[ y := y - 1; \]
\[ z := z - 1; \]

one abstract transition - Single-path loop

several transitions - Multiple-path loop
Usefulness of constraints

A loop with truncating division

while (4*x >= y & y>=1) do x := (2*x+1)/5

Can be represented as:

while (4x >= y & y>=1) do
  5x' <= 2x + 1
  5x' >= 2x - 3
  y' = y
Single-path Linear Constraint Loops (over integers)

Widely studied

An important reason:

The “one loop at a time” approach to proving termination

(size-change termination, transition invariants, …)

An interesting question of decidability (open!!), partial results: Braverman CAV’06, BGM at VMCAI ’12
Linear Ranking Functions

• By example

• Here $f(x, y, z) = x + y$ is a ranking function
  - non-negative in all (enabled) states
  - strictly decreasing
  - proves termination
Linear Ranking Functions

• By example

\[
\text{while } (y \leq x \land x+y \geq 1) \text{ do } \\
x' = x \land y' = y+1-2x
\]

• Here \( f(x,y,z) = x+y \) is a ranking function
  - non-negative in all (enabled) states
  - strictly decreasing
  - proves termination
Linear Ranking Functions

• By example

\[
\text{while } (y \leq x \ \& \ x+y \geq 1) \ \text{do} \\
x' = x \ \& \ y' = y+1-2x
\]

• Here \( f(x,y,z) = x+y \) is a ranking function
  - non-negative in all (enabled) states
  - strictly decreasing
  - proves termination
Linear Ranking Functions

• By example

\[
\text{while } (y \leq x \land x+y \geq 1) \text{ do } \\
x' = x \land y' = y+1-2x
\]

• Here \( f(x,y,z) = x+y \) is a ranking function
  - non-negative in all (enabled) states
  - strictly decreasing
  - proves termination
  - for integers but not for rationals! (consider \( (\frac{1}{2}, \frac{1}{2}) \))
Uses of Linear Ranking Functions
Uses of Linear Ranking Functions

- Termination of imperative/etc. programs
Uses of Linear Ranking Functions

• Termination of imperative/etc. programs
• Complexity analysis (execution time, etc)
  - An integer-valued ranking function bounds the number of iterations / length of call chain
Uses of Linear Ranking Functions

- Termination of imperative/etc. programs
- Complexity analysis (execution time, etc)
  - An integer-valued ranking function bounds the number of iterations / length of call chain
- Loop parallelization
  - how to schedule computations that depend on previous results
Definitions
Definitions

- **State:** \( \vec{x} = (x_1, \ldots, x_n) \)
Definitions

- State: $\vec{x} = (x_1, \ldots, x_n)$
- Transition: $\vec{x}'' = (\vec{x}, \vec{x}') = (x_1, \ldots, x_n, x'_1, \ldots, x'_n)$
Definitons

• State: \( \bar{x} = (x_1, \ldots, x_n) \)

• Transition: \( \bar{x}'' = (\bar{x}, \bar{x}') = (x_1, \ldots, x_n, x'_1, \ldots, x'_n) \)

• An SLC loop is specified by a conjunction of transition constraints: \( A'' \bar{x}'' \leq c'' \)
  - this specifies a convex polyhedron \( Q \subseteq \mathbb{Q}^{2n} \) the transition polyhedron
  - the set of its integer points is \( I(Q) \)
Definitions

• State: \( \overrightarrow{x} = (x_1, \ldots, x_n) \)

• Transition: \( \overrightarrow{x}'' = (\overrightarrow{x}, \overrightarrow{x'}) = (x_1, \ldots, x_n, x'_1, \ldots, x'_n) \)

• An SLC loop is specified by a conjunction of transition constraints: \( A'' \overrightarrow{x}'' \leq c'' \)
  - this specifies a convex polyhedron \( Q \subset \mathbb{Q}^{2n} \)
  - the transition polyhedron
  - the set of its integer points is \( I(Q) \)

• \( f : \mathbb{Q}^n \rightarrow \mathbb{Q} \) is a ranking function (over rationals) if
  \( \overrightarrow{x}'' \in Q \Rightarrow f(\overrightarrow{x}) \geq 0 \wedge f(\overrightarrow{x}) - f(\overrightarrow{x'}) \geq 1 \)
Definitions

- State: \[ \vec{x} = (x_1, \ldots, x_n) \]
- Transition: \[ \vec{x}'' = (\vec{x}, \vec{x}') = (x_1, \ldots, x_n, x'_1, \ldots, x'_n) \]
- An SLC loop is specified by a conjunction of transition constraints: \[ A'' \vec{x}'' \leq c'' \]
  - this specifies a convex polyhedron \( Q \subset \mathbb{Q}^{2n} \)
  - the transition polyhedron
  - the set of its integer points is \( I(Q) \)
- \( f : \mathbb{Q}^n \to \mathbb{Q} \) is a ranking function (over rationals) if \( \vec{x}'' \in Q \Rightarrow f(\vec{x}) \geq 0 \land f(\vec{x}) - f(\vec{x}') \geq 1 \)
- It is a ranking function over the integers if the above condition holds for \( I(Q) \)
The Linear Ranking Problem

**Instance:** An SLC loop specified by a system of linear constraints $Q$

**Question:** Does there exist a LinRF for $Q$ ?
The Linear Ranking Problem

**Instance:** An SLC loop specified by a system of linear constraints $Q$

**Question:** Does there exist a LinRF for $Q$?

- Two variations of the problem:
  - Variables take **rational** values -- LinRF($\mathbb{Q}$)
  - Variables take **integer** values -- LinRF($\mathbb{Z}$)
Linear Ranking Functions

We are looking for a function \( f(\bar{x}) = \bar{\lambda} \cdot \bar{x} + \lambda_0 \) s.t.

\[
\begin{align*}
\bar{x}'' & \in Q \Rightarrow \bar{\lambda} \cdot \bar{x} + \lambda_0 \geq 0 \land \bar{\lambda} \cdot \bar{x} - \bar{\lambda} \cdot \bar{x}' \geq 1
\end{align*}
\]
Linear Ranking Functions

We are looking for a function $f(\vec{x}) = \vec{\lambda} \cdot \vec{x} + \lambda_0$ s.t.

$\vec{x}'' \in Q \Rightarrow \vec{\lambda} \cdot \vec{x} + \lambda_0 \geq 0 \land \vec{\lambda} \cdot \vec{x} - \vec{\lambda} \cdot \vec{x}' \geq 1$

(B) Linear constraints

(D) Linear Inequality
We are looking for a function $f(\vec{x}) = \vec{\lambda} \cdot \vec{x} + \lambda_0$ s.t.

(B) $\vec{x}'' \in Q \Rightarrow \vec{\lambda} \cdot \vec{x} + \lambda_0 \geq 0 \land \vec{\lambda} \cdot \vec{x} - \vec{\lambda} \cdot \vec{x}' \geq 1$

• suggests solution by Linear Programming: Using **Farkas’ Lemma**, we can synthesize inequalities implied by $Q$
LinRF by Linear Programming (LP)

- Sohn and van Gelder (1991)
- Feautrier (1992)
- Colón and Sipma (2001)
- Podelski and Rybalchenko (2004)
- Mesnard and Serebrenik (2008)
- Alias, Darte, Feautrier, Gonnord (2010)
Completeness

• We call a method complete if it is guaranteed to find a LinRF, when one exists.

• LP based methods are complete \((\text{PTIME})\)
Completeness

- We call a method \textit{complete} if it is guaranteed to find a LinRF, when one exists.
- LP based methods are complete \textbf{(PTIME)} over the rationals.
Completeness

• We call a method **complete** if it is guaranteed to find a LinRF, when one exists.

• LP based methods are complete **(PTIME)** over the rationals

• \((B)+(D)\) have to hold over \(\mathbb{Q}\), not just \(I(\mathbb{Q})\)
  
  - must fail for the two loops shown
Our work
Our work

• What is the complexity of the LinRF($\mathbb{Z}$) problem?
  - The decision problem is coNP-complete
  - contrast with PTIME for LinRF($\mathbb{Q}$)
  - we show:
    • synthesis algorithm in exponential time
    • coNP hardness
    • inclusion in coNP
    • some PTIME-solvable classes (DBMs? octagons?)
The source of hardness
The source of hardness
The source of hardness

polyehdron $Q$

integer hull = convex hull of $I(Q)$
Clue to solution
Clue to solution

- Since $Q_I$ it is integral:
Clue to solution

- Since $Q_I$ it is integral:
  - any rational point of $Q_I$ is a convex combination of some integer points $Q_I$
Clue to solution

• Since \( \mathbb{Q}_I \) it is integral:
  - any rational point of \( \mathbb{Q}_I \) is a convex combination of some integer points \( \mathbb{Q}_I \)
  - an inequality is implied by \( \mathbb{Q}_I \) over the rationals iff it is implied by \( \mathbb{Q}_I \) over the integers.
Clue to solution

- Since $\mathbb{Q}_I$ it is integral:
  - any rational point of $\mathbb{Q}_I$ is a convex combination of some integer points $\mathbb{Q}_I$
  - an inequality is implied by $\mathbb{Q}_I$ over the rationals \textbf{iff} it is implied by $\mathbb{Q}_I$ over the integers.
  - Conditions (B)+(D) are just inequalities
Clue to solution

• Since $Q_I$ it is integral:
  - any rational point of $Q_I$ is a convex combination of some integer points $Q_I$
  - an inequality is implied by $Q_I$ over the rationals iff it is implied by $Q_I$ over the integers.
  - Conditions (B)+(D) are just inequalities
  - $\text{LinRF}(\mathbb{Q})$ and $\text{LinRF}(\mathbb{Z})$ are equivalent
An algorithm for LinRF(\mathbb{Z})
An algorithm for LinRF($\mathbb{Z}$)

- Compute a constraints representation of the integer hull $Q_I$ of $Q$
An algorithm for LinRF(\(\mathbb{Z}\))

- Compute a constraints representation of the integer hull \(Q_I\) of \(Q\)
- Apply LinRF(\(\mathbb{Q}\)) on \(Q_I\): find LinRF over the rationals
An algorithm for LinRF(\(\mathbb{Z}\))

• Compute a constraints representation of the integer hull \(\mathbb{Q}_I\) of \(\mathbb{Q}\)
• Apply LinRF(\(\mathbb{Q}\)) on \(\mathbb{Q}_I\): find LinRF over the rationals
• Complete solution, exponential complexity
An algorithm for LinRF(\mathbb{Z})

- Compute a constraints representation of the integer hull \( Q_I \) of \( Q \)
- Apply LinRF(\mathbb{Q}) on \( Q_I \): find LinRF over the rationals
- Complete solution, exponential complexity
- The “exponential complexity” comes from computing the integer hull, LinRF(\mathbb{Q}) is PTIME
while (x₂ <= x₁ & x₁ + x₂ >= 1) do
  x₁' = x₁
  x₂' = x₂ + 1 - 2x₁

while \((x_2 \leq x_1 \& x_1 + x_2 \geq 1)\) do
\[ x_1' = x_1 \]
\[ x_2' = x_2 + 1 - 2x_1 \]

The loop \(Q\) has a LinRF over the integers but not over the rationals.
while \((x_2 \leq x_1 \& x_1 + x_2 \geq 1)\) do
\[
\begin{align*}
x_1' &= x_1 \\
x_2' &= x_2 + 1 - 2x_1
\end{align*}
\]

The loop \(Q\) has a LinRF over the integers but not over the rationals

The loop \(Q_I\) has a LinRF over the integers and the rationals
Hardness Proof for LinRF(\mathbb{Z})
Hardness Proof for LinRF(\(\mathbb{Z}\))

- NP-hard problem
  - Given constraints \(B\vec{x} \leq b\), is there an integer solution? (NP-Hard problem)
  - Reduction to the existence of LinRF

\[
\text{while } (B\vec{x} - \vec{z} \leq b \land \vec{z} \geq 0) \text{ do } \vec{x}' = \vec{x}, \vec{z}' = 0
\]
Hardness Proof for LinRF(\(\mathbb{Z}\))

- NP-hard problem
  - Given constraints \(B\bar{x} \leq b\), is there an integer solution? (NP-Hard problem)
  - Reduction to the existence of LinRF

\[
\text{while } (B\bar{x} - \bar{z} \leq b \& \bar{z} \geq 0) \text{ do } \bar{x}' = \bar{x}, \; \bar{z}' = 0
\]

If there is an integer solution \(\bar{x}_0\): non-terminating (\(\bar{z} = 0\))
Hardness Proof for LinRF(\mathbb{Z})

- NP-hard problem
  - Given constraints $B\vec{x} \leq b$, is there an integer solution? (NP-Hard problem)
  - Reduction to the existence of LinRF

\begin{equation}
\text{while } (B\vec{x} - \vec{z} \leq b \land \vec{z} \geq 0) \text{ do } \vec{x}' = \vec{x}, \vec{z}' = 0
\end{equation}

If there is an integer solution $\vec{x}_0$: non-terminating ($\vec{z} = 0$)

If there is an no integer solution, it has a LinRF $z_1 + \cdots + z_n$
Inclusion of LinRF($\mathbb{Z}$) in coNP

Theorem
The LinRF($\mathbb{Z}$) problem is coNP-complete.

Proof. We show that there is polynomially checkable witnesses to non-existence of a LinRF.
Definitions

Consider a candidate function \( f(\tilde{x}) = \tilde{\lambda} \cdot \tilde{x} + \lambda_0 \)

Point \( \tilde{x}'' = (\tilde{x}, \tilde{x}') \in \mathcal{Q} \) witnesses against \( f \) if \( f \) fails to satisfy (B) or (D)

\[
\begin{align*}
\tilde{\lambda} \cdot \tilde{x} + \lambda_0 &\geq 0 \quad \text{(B)} \\
\tilde{\lambda} \cdot \tilde{x} - \tilde{\lambda} \cdot \tilde{x}' &\geq 1 \quad \text{(D)}
\end{align*}
\]

Let \( \mathcal{W}(\tilde{x}'') = \{\tilde{\lambda}|\tilde{x}'' \text{ witnesses against } \tilde{\lambda}\} \)

\[
\mathcal{W}(X) = \bigcup_{x \in X} \mathcal{W}(\tilde{x}'')
\]

no LinRF \( \iff \exists X . \mathcal{W}(X) = \mathbb{Q}^{n+1} \)
Definitions

Consider a candidate function \( f(\tilde{x}) = \lambda \cdot \tilde{x} + \lambda_0 \)

Point \( \tilde{x}'' = (\tilde{x}, \tilde{x}') \in \mathcal{Q} \) witnesses against \( f \) if \( f \) fails to satisfy (B) or (D)

\[
\begin{align*}
\lambda \cdot \tilde{x} + \lambda_0 & \geq 0 \quad (B) \\
\lambda \cdot \tilde{x} - \lambda \cdot \tilde{x}' & \geq 1 \quad (D)
\end{align*}
\]

Let \( W(\tilde{x}'') = \{ \lambda \mid \tilde{x}'' \text{ witnesses against } \lambda \} \)

\[
W(X) = \bigcup_{x \in X} W(\tilde{x}'')
\]

no LinRF \( \iff \exists X . \ W(X) = \mathbb{Q}^{n+1} \)
• Where to look for witnesses?

1. If \( Q \) is bounded:

\[
Q = \text{conv. hull}(x_1, \ldots, x_V)
\]

If \( f(\vec{x}) \) is not a RF, it must fail on one of the vertices. (If all vertices satisfy (B),(D), then so does any convex combination.)

\[
\Rightarrow \text{no LinRF} \iff W(\{x_1, \ldots, x_V\}) = \mathbb{Q}^{n+1}
\]
• Where to look for witnesses?

1. If $Q$ is bounded:

$$Q = \text{conv. hull}(x_1, ..., x_V)$$

If $f(\bar{x})$ is not a RF, it must fail on one of the vertices. (If all vertices satisfy (B),(D), then so does any convex combination.)

$$\Rightarrow \text{no LinRF} \iff W(\{x_1, ..., x_V\}) = Q^{n+1}$$
• Where to look for witnesses?

1. If $Q$ is bounded:

$$Q = \text{conv. hull}(x_1, ..., x_V)$$

If $f(x)$ is not a RF, it must fail on one of the vertices. (If all vertices satisfy (B),(D), then so does any convex combination.)

$$\Rightarrow \text{no LinRF} \iff W(\{x_1, ..., x_V\}) = \mathbb{Q}^{n+1}$$
• Where to look for witnesses?

\[ \text{no LinRF } \iff WS(x_1, \ldots, x_V) = \mathbb{Q}^{n+1} \]
\[ \iff (B),(D) \text{ fail on } x_1, \ldots, x_V \]

A corollary of Farkas' Lemma (found in Schrijver): “if a set of linear constraints on \( \mathbb{Q}^{n+1} \) has no solution, there is a subset of \( n+2 \) of them that doesn’t”

- We have a small witness set
- The bit-size of the witnesses is polynomial
  (theorem on relation between bit-size of constraints and of vertices.)
• Unbounded polyhedra

Every polyhedron can be represented as

\[ Q = \text{conv. hull}(x_1, \ldots, x_V) + \text{cone}(y_1, \ldots, y_R) \]

If \( f(\bar{x}) \) is not a RF, it must fail on one of the vertices, or on one of the rays.

A ray \( y \) is added to a point \( x \in \text{conv. hull}(x_1, \ldots, x_V) \) to form points \( x + ay, \ a \geq 0 \)

\( y \) witnesses against \( f \) if \( f \) fails to satisfy (B') or (D')

\[ \lambda \cdot \bar{y} \geq 0 \quad \text{(B')} \quad \lambda \cdot \bar{y} - \lambda \cdot \bar{y}' \geq 0 \quad \text{(D')} \]
Cases we can solve in polynomial time

Our algorithm for LinRF(\(\mathbb{Z}\)):

- Compute the integer hull \(Q_I\) (EXPTIME)
- Apply LinRF(\(Q\)) on \(Q_I\) (PTIME)

Our problem becomes tractable if either:

- \(Q\) is an integral polyhedron (\(Q_I = Q\)) [CKRW'10]
- we have a specialized procedure to compute \(Q_I\)
Totally Unimodular matrices
Totally Unimodular matrices

Matrix $A$ is totally unimodular if each subdeterminant of $A$ is in $\{0, \pm 1\}$. $A$ is TUM $\Rightarrow$ the polyhedron \{x|Ax \leq b\} is integral.

Example:

Difference-bound constraints yield TUM constraint matrices.

\[ x_i - x_j \leq d \]
\[ \pm x_i \leq d \]
Difference-bound constraints are special cases of:

- Octagons

\[ \pm x_i \pm x_j \leq d \]
\[ \pm x_i \leq d \]

- Two-variable per inequality (TVPI) constraints

\[ ax_i + bx_j \leq d \]
Octagons
Octagons can be non-integral. **Tight closure** is used to test satisfiability. Basically it adds constraints...
Octagons

Octagons can be non-integral. **Tight closure** is used to test satisfiability. Basically it adds constraints
Octagons

Octagons can be non-integral. **Tight closure** is used to test satisfiability. Basically it adds constraints.
Octagons

Octagons can be non-integral. **Tight closure** is used to test satisfiability. Basically it adds constraints.
Octagons

Octagons can be non-integral. **Tight closure** is used to test satisfiability. Basically it adds constraints

But in $\geq 3$ dimensions, the result can be non-integral

\[
\begin{align*}
    x_1 + x_2 &\leq 2 \\
    x_1 + x_3 &\leq 3 \\
    x_2 + x_3 &\leq 4
\end{align*}
\]
Octagons

Octagons can be non-integral. Tight closure is used to test satisfiability. Basically it adds constraints

But in $\geq 3$ dimensions, the result can be non-integral

while($x_1+x_2 \leq 2$ \& $x_1+x_3 \leq 3$ \& $x_2+x_3 \leq 4$) do
  $x_1' = 1-x_1$
  $x_2' = 1+x_1$
  $x_3' = 1+x_2$
Octagons

Octagons can be non-integral. **Tight closure** is used to test satisfiability. Basically it adds constraints

But in ≥3 dimensions, the result can be non-integral

while(x1+x2 <= 2 & x1+x3 <= 3 & x2+x3 <= 4) do
  x1' = 1-x1
  x2' = 1+x1
  x3' = 1+x2

The LinRF(\(\mathbb{Z}\)) problem is coNP-hard if the guard is an octagons, and the update is linear!
Two variables per inequality

Harvey (1999) show how to compute in PTIME the integer hull of any two-dimensional polyhedron.

More than 2 variables? Try to decompose the constraint set into independent constraint-sets.

```
while (4x >= 1 & y>=1) do
    5x' <= 2x + 1,
    5x' > 2x - 4,
    y = y'
```
Two variables per inequality

Harvey (1999) show how to compute in PTIME the integer hull of any \textit{two-dimensional} polyhedron.

More than 2 variables? Try to \textit{decompose} the constraint set into independent constraint-sets.

\begin{verbatim}
while (4x >= 1 & y>=1) do
  5x' <= 2x + 1,
  5x' > 2x - 4,
  y = y'
\end{verbatim}
Two variables per inequality

Harvey (1999) show how to compute in PTIME the integer hull of any two-dimensional polyhedron.

More than 2 variables? Try to decompose the constraint set into independent constraint-sets.

while \((4x \geq 1 \& y \geq 1)\) do
\[
\begin{align*}
5x' & \leq 2x + 1, \\
5x' & > 2x - 4, \\
y & = y'
\end{align*}
\]
while (-x+y ≤ 0 & -2x-y ≤ -1 & z ≤ 1) do
    x' = x,
    y' = y - 2x + z,
    z' = z
while \((-x + y \leq 0 \& -2x - y \leq -1 \& z \leq 1\) do

\[\begin{align*}
x' &= x, \\
y' &= y - 2x + z, \\
z' &= z
\end{align*}\]

- Decomposition fails

**THM:** if the *guard* is an integral polyhedron, and the *update* is linear \((\tilde{x}' = A\tilde{x} + \tilde{b})\), then the transition polyhedron is integral.

- Above: update is linear

  guard solved by decomposition
Summary
Summary

- Linear ranking functions are often used, and often over integers
Summary

• Linear ranking functions are often used, and often over integers
• The decision problem is coNP-complete
Summary

- Linear ranking functions are often used, and often over integers
- The decision problem is coNP-complete
- Can be solved by exponential algorithm, in general
Summary

- Linear ranking functions are often used, and often over integers
- The decision problem is coNP-complete
- Can be solved by exponential algorithm, in general
- Or by polynomial algorithms, for certain classes of constraints
Summary

• Linear ranking functions are often used, and often over integers
• The decision problem is coNP-complete
• Can be solved by exponential algorithm, in general
• or by polynomial algorithms, for certain classes of constraints
• Practically, the solution over rationals can always be tried before an exponential algorithm
Final remarks

- Our result also provide a polynomial upper-bound on the bit-size of $\vec{\lambda}, \lambda_0$

- Bradely, Manna, Sipma (CONCUR’05)
  - solve the LinRF problem for integer data
  - use recursive “bisection” search over rationals - no termination guaranteed
  - we can supply the bounds to stop it
http://www.loopkiller.com/irankfinder