# Verified Linear Programming

Christian Keil

Institute for Reliable Computing
Hamburg University of Technology

GICOLAG, 2006

# Outline


1. Introduction
2. Rigorous Bounds
3. Numerical experience
   - Netlib
   - Comparison
4. Software
5. Summary

# Outline


1. Introduction
2. Rigorous Bounds
3. Numerical experience
   - Netlib
   - Comparison
4. Software
5. Summary

# Why linear programming?

- most basic form of optimization
- wide range of applications:
  oil refinery problems, flap settings on aircraft, industrial production and allocation, image restoration, linearization, linear relaxations in global optimization, ...

**Lovasz (1980)**

*If one would take statistics about which mathematical problem is using up most of the computer time in the world, then (not including database handling problems like sorting and searching) the answer would probably be linear programming.*

# And why verified?

- Ordóñez and Freund (2003):
  71% of Netlib lp problems are ill-posed

**Ben-Tal and Nemirovski (2000)**

*In real-world applications of Linear Programming one cannot ignore the possibility that a small uncertainty in the data (intrinsic for most real-world LP programs) can make the usual optimal solution of the problem completely meaningless from a practical viewpoint.*

# A representation of a linear program

## Definition (Linear Program)

Find the optimal value $f^*$ of a linear objective function $c^T x$ subject to

- linear constraints $Ax \leq a,\ Bx = b$ and
- simple bounds $\underline{x} \leq x \leq \overline{x}$.

- Set of feasible points $F$ satisfying constraints and simple bounds
- Can be represented by the tuple $P := (c, A, a, B, b)$ and $\underline{x}, \overline{x}$
- Simple bounds may be infinite

# Outline


1. Introduction
2. Rigorous Bounds
3. Numerical experience
   - Netlib
   - Comparison
4. Software
5. Summary

# A lower bound for the optimal value (1)

**Theorem (Jansson (2004))**

*Given a linear program P and $\underline{x}, \overline{x}$. If $\mathbf{y} \leq 0, \mathbf{z}$ satisfy*

- $\exists\, y \in \mathbf{y}, z \in \mathbf{z} : c_j - (A_{:j})^T y - (B_{:j})^T z = 0$ *for free* $x_j$, *and*
- *the defects*

$$\mathbf{d}_j := c_j - (A_{:j})^T \mathbf{y} - (B_{:j})^T \mathbf{z} \begin{cases} \leq 0 & \text{for} \quad -\infty < x_j \leq \overline{x}_j \\ \geq 0 & \text{for} \quad \underline{x}_j \leq x_j < \infty \end{cases}$$

*then a lower bound for the optimal value is*

$$\underline{f}^* := \inf\{a^T \mathbf{y} + b^T \mathbf{z} + \sum_{\underline{x}_j \neq -\infty} \underline{x}_j \mathbf{d}_j^+ + \sum_{\overline{x}_j \neq \infty} \overline{x}_j \mathbf{d}_j^- \}.$$

# A lower bound for the optimal value (2)

- $O(n^2)$ operations for finite simple bounds, coincides with Neumaier and Shcherbina's (2004)
- Conditions not met (infinite simple bounds) $\Rightarrow$ iterate with perturbed dual constraints
- Works for interval problems **P**

# An upper bound for the optimal value

- Based on idea from Krawczyk (1975) later used and modified
  Jansson (1988), Hansen and Walster (1991), Kearfott (1994)
- Enclose primal interior point
- Verifies existence of primal feasible solutions
- Works for interval problems **P**

# Outline


1. Introduction
2. Rigorous Bounds
3. Numerical experience
   - Netlib
   - Comparison
4. Software
5. Summary

# A real world application test set

Netlib comprises $\sim$ 100 problems

- First added in 1988, latest in 1996
- 32 to 15695 variables, 27 to 16675 constraints (medium size)
- Established test set for lp algorithms
- Ordóñez and Freund (2003): 71% ill-conditioned

# Condition and the distance to infeasibility

## Definition (Distance to Infeasibility $\rho$)

Norm of the smallest perturbation of $P$
that results in an empty feasible set.

$\rho_P$ distance to primal infeasibility

$\rho_D$ distance to dual infeasibility

## Definition (Condition)

Scale invariant reciprocal of the minimal distance to infeasibility.

$$C(d) := \frac{\|d\|}{\min\{\rho_P, \rho_D\}}$$

# Some examples of what can happen

| | $\rho_D$ | $\rho_P$ | $\underline{f}^*$ | $\overline{f}^*$ | $\mu$ |
|---|---|---|---|---|---|
| scagr25 | 0.034646 | 0.021077 | √ | √ | $3.7821e-8$ |
| bnl1 | 0.106400 | 0 | √ | | $7.2244e-8$ |
| stair | 0 | 0.000580 | | √ | $5.4796e-9$ |
| 80bau3b | 0 | 0 | √ | √ | $5.6653e-8$ |
| scsd8 | 1.000000 | 0.268363 | √ | | $6.3831e-8$ |

| | m | p | n | nz |
|---|---|---|---|---|
| scagr25 | 171 | 300 | 500 | 1554 |
| bnl1 | 411 | 232 | 1175 | 5121 |
| stair | 147 | 209 | 467 | 3856 |
| 80bau3b | 2262 | 0 | 9799 | 21002 |
| scsd8 | 0 | 397 | 2750 | 8584 |

# Some examples of what can happen

| | $\rho_D$ | $\rho_P$ | $\underline{f}^*$ | $\overline{f}^*$ | $\mu$ |
|---|---|---|---|---|---|
| scagr25 | 0.034646 | 0.021077 | $\surd$ | $\surd$ | $3.7821e-8$ |
| bnl1 | 0.106400 | 0 | $\surd$ | | $7.2244e-8$ |
| stair | 0 | 0.000580 | | $\surd$ | $5.4796e-9$ |
| 80bau3b | 0 | 0 | $\surd$ | $\surd$ | $5.6653e-8$ |
| scsd8 | 1.000000 | 0.268363 | $\surd$ | | $6.3831e-8$ |

| | m | p | n | nz |
|---|---|---|---|---|
| scagr25 | 171 | 300 | 500 | 1554 |
| bnl1 | 411 | 232 | 1175 | 5121 |
| stair | 147 | 209 | 467 | 3856 |
| 80bau3b | 2262 | 0 | 9799 | 21002 |
| scsd8 | 0 | 397 | 2750 | 8584 |

# Some examples of what can happen

| | $\rho_D$ | $\rho_P$ | $\underline{f}^*$ | $\overline{f}^*$ | $\mu$ |
|---|---|---|---|---|---|
| scagr25 | 0.034646 | 0.021077 | $\surd$ | $\surd$ | $3.7821e-8$ |
| bnl1 | 0.106400 | 0 | $\surd$ | | $7.2244e-8$ |
| stair | 0 | 0.000580 | | $\surd$ | $5.4796e-9$ |
| 80bau3b | 0 | 0 | $\surd$ | $\surd$ | $5.6653e-8$ |
| scsd8 | 1.000000 | 0.268363 | $\surd$ | | $6.3831e-8$ |

| | m | p | n | nz |
|---|---|---|---|---|
| scagr25 | 171 | 300 | 500 | 1554 |
| bnl1 | 411 | 232 | 1175 | 5121 |
| stair | 147 | 209 | 467 | 3856 |
| 80bau3b | 2262 | 0 | 9799 | 21002 |
| scsd8 | 0 | 397 | 2750 | 8584 |

# Overview of the results

In total 89 problems, 86 with infinite simple bounds

- 35 finite upper bounds for the optimal value

$$\text{med}(\mu(\overline{f}^*, f^*)) = 8.0e-9 \qquad \text{med}(t_{\overline{f}^*}/t_{f^*}) = 5.3$$

- 76 finite lower bounds for the optimal value

$$\text{med}(\mu(\underline{f}^*, f^*)) = 2.2e-8 \qquad \text{med}(t_{\underline{f}^*}/t_{f^*}) = 0.5$$

# Software to solve LP rigorously

- iCOs, RealPaver
  - constraint satisfaction codes
  - no optimization $\Rightarrow$ lower bound from non-satisfiable problems
  - iCOs: constraint programming, interval analysis
  - RealPaver: branch-and-prune, interval Newton method

# Software to solve LP rigorously

- iCOs, RealPaver
  - constraint satisfaction codes
  - no optimization $\Rightarrow$ lower bound from non-satisfiable problems
  - iCOs: constraint programming, interval analysis
  - RealPaver: branch-and-prune, interval Newton method
- perPlex
  - verification tool using rational arithmetic
  - checks feasibility and optimality of a (approximate) basis
  - requires rational solution
  - cannot handle interval data
  - proof of concept (cannot compute solution, proceed from suboptimal basis)

# More software to solve LP rigorously

- GlobSol, Numerica, COSY
  - global optimization
  - GlobSol: interval branch-and-bound, automatic differentiation, constraint propagation, interval Newton, ...
  - Numerica: interval methods (Hansen–Sengupta), constraint satisfaction
  - COSY: branch-and-bound, Taylor model arithmetic

# More software to solve LP rigorously

- GlobSol, Numerica
  - global optimization
  - GlobSol: interval branch-and-bound, automatic differentiation, constraint propagation, interval Newton, ...
  - Numerica: interval methods (Hansen–Sengupta), constraint satisfaction

# More software to solve LP rigorously

- GlobSol, Numerica
  - global optimization
  - GlobSol: interval branch-and-bound, automatic differentiation, constraint propagation, interval Newton, ...
  - Numerica: interval methods (Hansen–Sengupta), constraint satisfaction
- Lurupa
  - verified linear programming in the presence of uncertainty

# A proper test set

- 99 random problems after Rosen and Suzuki
  - choose solution $\Rightarrow$ generate problem from KKT
  - non–degenerate
  - exactly known, integral solution
  - 3 instances per dimension set (variables, inequalities, equations)
- 19 real-world problems from Netlib and Meszaros's collection
  - less than 50 variables
  - large finite search region
- 1 hour timeout

# Results for random problems

5 variables (total 36)

| | solved | $t_{max}$ |
|---|---|---|
| iCOs | 36 | 0.091s |
| RealPaver | 34 | 1378.100s |
| perPlex | 36 | 0.028s |
| GlobSol | 36 | 548.582s |
| Lurupa | 36 | < 0.010s |

Numerica solved 5 inequalitites and 5 variables in 326.7s

# Results for random problems

5 variables (total 36)

| | solved | $t_{max}$ |
|---|---|---|
| iCOs | 36 | 0.091s |
| RealPaver | 34 | 1378.100s |
| perPlex | 36 | 0.028s |
| GlobSol | 36 | 548.582s |
| Lurupa | 36 | < 0.010s |

Numerica solved 5 inequalitites and 5 variables in 326.7s

# Results for random problems

5 variables (total 36)

| | solved | $t_{max}$ |
|---|---|---|
| iCOs | 36 | 0.091s |
| RealPaver | 34 | 1378.100s |
| perPlex | 36 | 0.028s |
| GlobSol | 36 | 548.582s |
| Lurupa | 36 | < 0.010s |

Numerica solved 5 inequalitites and 5 variables in 326.7s

10 variables (total 36)

| | solved | $t_{max}$ |
|---|---|---|
| iCOs | 36 | 0.254s |
| RealPaver | 0 | |
| perPlex | 36 | 0.036s |
| GlobSol | 3 | 666.963s |
| Lurupa | 36 | < 0.010s |

# Results for larger random problems

25 variables (total 18)

| | solved | $t_{max}$ |
|---|---|---|
| iCOs | 5 | 0.273s |
| perPlex | 18 | 0.077s |
| Lurupa | 18 | $<$ 0.010s |

iCOs

- solved problems with 15 or 25 equations
- ran out of memory for the remaining problems

# Results for larger random problems

25 variables (total 18)

| | solved | $t_{max}$ |
|---|---|---|
| iCOs | 5 | 0.273s |
| perPlex | 18 | 0.077s |
| Lurupa | 18 | < 0.010s |

iCOs

- solved problems with 15 or 25 equations
- ran out of memory for the remaining problems

500 variables (total 9)

| | solved | $t_{max}$ |
|---|---|---|
| perPlex | 0 | |
| Lurupa | 9 | 29.2s |

# Results for real-world problems

feasible problems
(total 15)

| | solved | $t_{max}$ |
|---|---|---|
| iCOs | 7 | 0.401s |
| RealPaver | 12 | 207.500s |
| perPlex | 15 | 0.021s |
| GlobSol | 6 | 4.564s |
| Lurupa | 15 | < 0.010s |

iCOs wronlgy claims 3 problems to be infeasible

# Results for real-world problems

feasible problems (total 15)

| | solved | $t_{max}$ |
|---|---|---|
| iCOs | 7 | 0.401s |
| RealPaver | 12 | 207.500s |
| perPlex | 15 | 0.021s |
| GlobSol | 6 | 4.564s |
| Lurupa | 15 | < 0.010s |

iCOs wronlgy claims 3 problems to be infeasible

infeasible problems (total 4)

| | terminated | $t_{max}$ |
|---|---|---|
| iCOs | 4 | 1.190s |
| RealPaver | 4 | < 0.010s |
| perPlex | 0 | |
| GlobSol | 0 | |
| Lurupa | 4 | 0.040s |

# Results for real-world problems

feasible problems (total 15)

| | solved | $t_{max}$ |
|---|---|---|
| iCOs | 7 | 0.401s |
| RealPaver | 12 | 207.500s |
| perPlex | 15 | 0.021s |
| GlobSol | 6 | 4.564s |
| Lurupa | 15 | < 0.010s |

iCOs wronlgy claims 3 problems to be infeasible

infeasible problems (total 4)

| | terminated | $t_{max}$ |
|---|---|---|
| iCOs | 4 | 1.190s |
| RealPaver | 4 | < 0.010s |
| perPlex | 0 | |
| GlobSol | 0 | |
| Lurupa | 4 | 0.040s |

# Results for real-world problems

feasible problems (total 15)

| | solved | $t_{max}$ |
|---|---|---|
| iCOs | 7 | 0.401s |
| RealPaver | 12 | 207.500s |
| perPlex | 15 | 0.021s |
| GlobSol | 6 | 4.564s |
| Lurupa | 15 | < 0.010s |

iCOs wronlgy claims 3 problems to be infeasible

infeasible problems (total 4)

| | terminated | $t_{max}$ |
|---|---|---|
| iCOs | 4 | 1.190s |
| RealPaver | 4 | < 0.010s |
| perPlex | 0 | |
| GlobSol | 0 | |
| Lurupa | 4 | 0.040s |

# Results for real-world problems

feasible problems (total 15)

| | solved | $t_{max}$ |
|---|---|---|
| iCOs | 7 | 0.401s |
| RealPaver | 12 | 207.500s |
| perPlex | 15 | 0.021s |
| GlobSol | 6 | 4.564s |
| Lurupa | 15 | < 0.010s |

iCOs wronlgy claims 3 problems to be infeasible

infeasible problems (total 4)

| | terminated | $t_{max}$ |
|---|---|---|
| iCOs | 4 | 1.190s |
| RealPaver | 4 | < 0.010s |
| perPlex | 0 | |
| GlobSol | 0 | |
| Lurupa | 4 | 0.040s |

Lurupa returns some wide bounds ($[0, \infty]$, $[-\infty, \infty]$)

# Outline

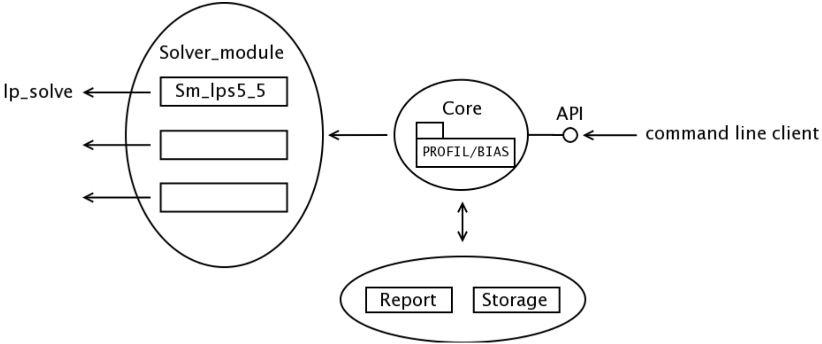

1. Introduction
2. Rigorous Bounds
3. Numerical experience
   - Netlib
   - Comparison
4. Software
5. Summary

# Lurupa's design goals

- Rigorous bounds for the optimal value
- Rigorous enclosure of near optimal, feasible points
- Use unmodified solvers plus postprocessing
- Easy to use
- Easily adoptable to different solvers
- Standalone and library version

# Lurupa's internals

- Implemented in ANSI C++
- Builds on interval library PROFIL/BIAS by Knüppel (1994)



# Usage of Lurupa

## Example (command line)

```
>lurupa -sm /path/to/module \
        -lp /path/to/model -lb -ub
```

Meaning of parameters

- -sm specify solver module
- -lp specify lp model file
- -lb compute lower bound
- -ub compute upper bound

# Usage of Lurupa

## Example (API)

```
Lurupa l;
l.set_solver_module("path/to/module");

FILE *in = fopen("path/to/model", "r");
Lp lp = l.read_lp(in, 0);

double optimal, bound, iterations;
l.solve_lp(lp, optimal);
l.lower_bound(lp, bound, iterations);
l.upper_bound(lp, bound, iterations);
```

# Outline


1. Introduction
2. Rigorous Bounds
3. Numerical experience
   - Netlib
   - Comparison
4. Software
5. Summary

# Summary

- Lurupa implements rigorous bounds and solves medium-scale problems in reasonable time
- General optimization software can verify solutions up to $\sim 10 - 20$ variables
- Larger dimensions require to exploit special structure

# Future work

- Verified condition measures
- Sparse structures (in PROFIL)
- New solver modules
- Certificates of infeasibility, unboundedness
- Verified preprocessing as suggested by Fourer and Gay (1993)

# Thank you for your

a- 

# Dimensions for real-world problems

Meszaros's misc problems

| | m | p | n |
|---|---|---|---|
| kleemin3 | 3 | 0 | 3 |
| kleemin4 | 4 | 0 | 4 |
| kleemin5 | 5 | 0 | 5 |
| kleemin6 | 6 | 0 | 6 |
| kleemin7 | 7 | 0 | 7 |
| kleemin8 | 8 | 0 | 8 |
| farm | 5 | 2 | 12 |
| p0033 | 15 | 0 | 33 |
| refine | 29 | 0 | 33 |
| p0040 | 23 | 0 | 40 |
| problem | 0 | 12 | 46 |

Meszaros's infeas problems

| | m | p | n |
|---|---|---|---|
| itest2 | 9 | 0 | 4 |
| galenet | 6 | 2 | 8 |
| itest6 | 9 | 2 | 8 |
| bgprtr | 6 | 14 | 35 |

Netlib problems

| | m | p | n |
|---|---|---|---|
| afiro | 19 | 8 | 32 |
| kb2 | 27 | 16 | 41 |
| sc50a | 30 | 20 | 48 |
| sc50b | 30 | 20 | 48 |

# Detailed results for real-world problems

Meszaros's misc problems

| | iCOs | | RealPaver | | perPlex | GlobSol | Lurupa |
|---|---|---|---|---|---|---|---|
| | t | $\Delta$ | t | $\Delta$ | | | |
| kleemin3 | 0.004 | 1e-12 | 0 | 1e-12 | 0.011 | 0.032 | < 0.010 |
| kleemin4 | 0.007 | 1e-12 | 0.030 | 1e-12 | 0.012 | 0.089 | < 0.010 |
| kleemin5 | 0.009 | 1e-12 | 0.210 | 1e-12 | 0.014 | 0.240 | < 0.010 |
| kleemin6 | ? | | 3.030 | 1e-12 | 0.012 | 0.639 | < 0.010 |
| kleemin7 | ? | | 207.500 | 1e-12 | 0.011 | 1.675 | < 0.010 |
| kleemin8 | ? | | 10.380 | 1e-4 | 0.012 | 4.564 | < 0.010 |
| farm | 0.009 | 1e-12 | 1.480 | 1e-8 | 0.012 | t | < 0.010 |
| p0033 | oom | | t | 1e-1 | 0.017 | t | < 0.010 |
| refine | oom | | 0 | 1e-1 | 0.021 | e/t | < 0.010 |
| p0040 | oom | | t | 1e-1 | 0.014 | t | < 0.010 |
| problem | 0.019 | 1e-12 | 0 | 1e-12 | 0.015 | t | < 0.010 |

# More detailed results for real-world problems

Meszaros's infeas problems

| | iCOs | RealPaver | perPlex | GlobSol | Lurupa |
|---|---|---|---|---|---|
| itest2 | 0.947 | 0 | lpinf | e/t | $<$ 0.010s $[0, \infty]$ |
| galenet | 0 | 0 | lpinf | e | $<$ 0.010s $[0, \infty]$ |
| itest6 | 1.190 | 0 | lpinf | e/t | $<$ 0.010s $[1.9e6, \infty]$ |
| bgprtr | 0.025 | 0 | lpinf | t | 0.040s $[-\infty, +\infty]$ |

Netlib problems

| | iCOs | | RealPaver | | perPlex | GlobSol | Lurupa |
|---|---|---|---|---|---|---|---|
| | t | $\Delta$ | t | $\Delta$ | | | |
| afiro | oom | | 55.680 | 1e-12 | 0.012 | t | $<$ 0.01 |
| kb2 | oom | | t | 1e-1 | 0.016 | t | $<$ 0.01 |
| sc50a | 0.401 | 1e-12 | 157.700 | 1e-2 | 0.016 | t | 0.01 |
| sc50b | 0.342 | 1e-12 | 0.740 | 1e-1 | 0.016 | t | $<$ 0.01 |