Using decomposition to address degeneracy in the dual simplex algorithm

Ambros Gleixner, Stephen J Maher, Matthias Miltenberger

Zuse Institute Berlin
Berlin, Germany

15th June 2015
Algorithmic linear programming

1947  Dantzig: primal **simplex algorithm**
1954  Lemke and Beale: dual simplex algorithm
1979  Khachiyan: **ellipsoid method**
1984  Karmarkar: primal interior point/barrier algorithm
1989  Kojima et al: primal-dual **interior point algorithm**
Simplex method - efficient and effective

- Effective approach to solve linear programming problems.
- Fundamental for solving mixed integer programs.
- Algorithm developments have greatly improved efficiency.
  - Linear algebra,
  - Pricing and pivoting rules,
  - Presolving techniques,
  - ...
Why improve the simplex method?

All about algorithm efficiency

- There are many features of the simplex methods that impact efficiency.
- One such feature is primal and dual degeneracy.
All about algorithm efficiency

- There are many features of the simplex methods that impact efficiency.
- One such feature is primal and dual degeneracy.

**Goal:** Reduce the degeneracy encountered in the simplex method to improve algorithm performance.
Outline

Introduction

Decomposition approach for the dual simplex

Solution approach and implementation

Results

Conclusion
Primal and dual degeneracy

Consider \( P = \{ x \in \mathbb{R} | Ax = b, x \geq 0 \} \) and let \( x^* \) be a basic solution.

**Primal Degeneracy**

\( x^* \) is degenerate if more than \( n - m \) elements are zero.

**Dual Degeneracy**

Degeneracy occurs if more than \( m \) primal variables have a reduced cost of zero.
Primal and dual degeneracy

Consider $P = \{ x \in \mathbb{R} | Ax = b, x \geq 0 \}$ and let $x^*$ be a basic solution.

**Primal Degeneracy**

$x^*$ is degenerate if more than $n - m$ elements are zero.

**Dual Degeneracy**

Degeneracy occurs if more than $m$ primal variables have a reduced cost of zero.
Consider $P = \{x \in \mathbb{R} | Ax = b, x \geq 0\}$ and let $x^*$ be a basic solution.

**Primal Degeneracy**

$x^*$ is degenerate if more than $n - m$ elements are zero.

**Dual Degeneracy**

Degeneracy occurs if more than $m$ primal variables have a reduced cost of zero.
Prevalence of dual degeneracy

The number of instances with a given degeneracy level - 927 instances

Collection of 1,226 LP instances - 299 have a degeneracy level less than 1%.
Approaches to address degeneracy

Primal degeneracy

Improved Primal Simplex - Elhallaoui et al. (2011)

- Developed with a focus on set partitioning problems.
- Decomposes problem to contain only non-degenerate primal variables.
- Further developments have resulted in an efficient solution approach.
Primal degeneracy

Improved Primal Simplex - Elhallaoui et al. (2011)

- Developed with a focus on set partitioning problems.
- Decomposes problem to contain only non-degenerate primal variables.
- Further developments have resulted in an efficient solution approach.
Approaches to address degeneracy

Primal degeneracy

Improved Primal Simplex - Elhallaoui et al. (2011)

- Developed with a focus on set partitioning problems.
- Decomposes problem to contain only non-degenerate primal variables.
- Further developments have resulted in an efficient solution approach.
Approaches to address degeneracy

Dual degeneracy

- Perturbation
  - Modification of the objective function coefficients
- Bound flipping ratio test
  - Changing variable from one bound to another without affecting dual feasibility.
  - Larger steps are afforded.
Introduction

Decomposition approach for the dual simplex

Solution approach and implementation

Results

Conclusion
Consider $P = \{ x \in \mathbb{R} | Ax = b, x \geq 0 \}$ and let $x^*$ be a basic solution.

Degeneracy occurs if more than $m$ primal variables have a reduced cost of zero.
Reducing dual degeneracy

Consider $P = \{x \in \mathbb{R} | Ax = b, x \geq 0\}$ and let $x^*$ be a basic solution. Degeneracy occurs if more than $m$ primal variables have a reduced cost of zero.
Reducing dual degeneracy

Consider $P = \{ \mathbf{x} \in \mathbb{R} | A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0 \}$ and let $\mathbf{x}^*$ be a basic solution.

Degeneracy occurs if more than $m$ primal variables have a reduced cost of zero.
Reducing dual degeneracy

Consider $P = \{ x \in \mathbb{R} | Ax = b, x \geq 0 \}$ and let $x^*$ be a basic solution.

Degeneracy occurs if more than $m$ primal variables have a reduced cost of zero.
Reducing dual degeneracy

Consider $P = \{x \in \mathbb{R}| Ax = b, x \geq 0 \}$ and let $x^*$ be a basic solution. Degeneracy occurs if more than $m$ primal variables have a reduced cost of zero.
Row form basis matrix

Given $P = \max \left\{ c'x : \left( \begin{array}{c} -I \\ A \end{array} \right) x \leq \begin{pmatrix} 0 \\ b \end{pmatrix} \right\}$.

- The row basis is an $n \times n$ matrix.
- Rows represent the constraints/variable bounds that are tight at the basic solution.
Row form basis matrix

Given $P = \max \left\{ c'x : \begin{pmatrix} -I \\ A \end{pmatrix} x \leq \begin{pmatrix} 0 \\ b \end{pmatrix} \right\}$.

- The row basis is an $n \times n$ matrix.
- Rows represent the constraints/variable bounds that are tight at the basic solution.
Row form basis matrix

Given \( P = \max \left\{ c'x : \begin{pmatrix} -I \\ A \end{pmatrix} x \leq \begin{pmatrix} 0 \\ b \end{pmatrix} \right\} \).

- The row basis is an \( n \times n \) matrix.
- Rows represent the constraints/variable bounds that are tight at the basic solution.
Formulation

Given $P = \max\{c'x : Ax \leq b\}$, and a row form basis $B$ ($Q = B^{-1}$).

- Partition the indices of $B$ by the dual multipliers.
  - $\mathcal{P}$ where $y_i > 0$ and $\mathcal{N}$ where $y_i = 0$.
- Reorder basis: $Q = (Q_{\mathcal{P}} | Q_{\mathcal{N}})$.
- Substitute $x = Q\bar{x}$, where $x = Q_{\mathcal{P}}\bar{x}_{\mathcal{P}} + Q_{\mathcal{N}}\bar{x}_{\mathcal{N}}$.
- Compatible rows identified by $A_i.Q_{\mathcal{N}} = 0$, $i \in \{1, \ldots, m\}$.
Formulation

Given \( P = \max \{ c' x : Ax \leq b \} \), and a row form basis \( B (Q = B^{-1}) \).

- Partition the indices of \( B \) by the dual multipliers.
  - \( \mathcal{P} \) where \( y_i > 0 \) and \( \mathcal{N} \) where \( y_i = 0 \).
- Reorder basis: \( Q = (Q_{\mathcal{P}} | Q_{\mathcal{N}}) \).
- Substitute \( x = Q \bar{x} \), where \( x = Q_{\mathcal{P}} \bar{x}_{\mathcal{P}} + Q_{\mathcal{N}} \bar{x}_{\mathcal{N}} \).
- Compatible rows identified by \( A_i. Q_{\mathcal{N}} = 0, i \in \{1, \ldots, m\} \).

\[
\begin{align*}
\max \quad & (c' Q_{\mathcal{P}}) \bar{x}_{\mathcal{P}} + (c' Q_{\mathcal{N}}) \bar{x}_{\mathcal{N}} \\
\text{subject to} \quad & A_C. Q_{\mathcal{P}} \bar{x}_{\mathcal{P}} + A_C. Q_{\mathcal{N}} \bar{x}_{\mathcal{N}} \leq b_C, \\
& A_I. Q_{\mathcal{P}} \bar{x}_{\mathcal{P}} + A_I. Q_{\mathcal{N}} \bar{x}_{\mathcal{N}} \leq b_I.
\end{align*}
\]
Reduced problem

The reduced costs are: $c' Q = y_B$.
Given $c' Q.N = y_N = 0$ and $A_c Q.N = 0$. 
Reduced problem

The reduced costs are: \( c'Q = y_B \).
Given \( c'Q_N = y_N = 0 \) and \( A_C.Q_N = 0 \).

\[
\begin{align*}
\text{max} & \quad (c'Q_P)\bar{x}_P \\
\text{subject to} & \quad A_C.Q_P \bar{x}_P \leq b_C.
\end{align*}
\]
**Complementary problem**

Know: Constraints currently at bounds.
Know: Constraints currently at bounds.

\[ \begin{align*}
\text{max} & \quad s \\
\text{subject to} & \quad A_i x = b_i \quad \forall i \in \mathcal{P}^*, \\
& \quad A_i x + s \leq b_i \quad \forall i \in \mathcal{I}^*, \\
& \quad s \in \mathbb{R}. 
\end{align*} \]
Outline

Introduction

Decomposition approach for the dual simplex

Solution approach and implementation

Results

Conclusion
Algorithm

Solve original problem

Is basis degenerate?

no

Decompose problem

Solve reduced problem

Update reduced problem.

Update complementary problem

Solve complementary problem

Is s nonnegative?

yes

Optimal solution found

no
Algorithm

Solve original problem

Is basis degenerate?

- no
  - Decompose problem
  - Solve reduced problem
  - Update reduced problem.

- yes
  - Update complementary problem
  - Solve complementary problem
  - Is $s$ nonnegative?
    - yes
      - Optimal solution found
    - no
      - Must achieve an appropriate level of degeneracy.
Algorithm

1. Solve original problem
2. Is basis degenerate?
   - no
   - yes
     - Decompose problem
       - Solve reduced problem
       - Update reduced problem
       - Update complementary problem
       - Solve complementary problem
       - Is s nonnegative?
         - yes
         - no
           - Optimal solution found

Decomposition creates a reduced problem that eliminates current degeneracy
Algorithm

1. Solve original problem

   - Is basis degenerate?
     - no
       - Decompose problem
       - Update reduced problem
     - yes
       - Solve reduced problem
       - Update complementary problem
         - Solve complementary problem
         - Is s nonnegative?
           - yes
             - Complementary problem is updated with respect to the tight constraints.
           - no
             - Reduced problem is updated to include violated constraints.
           - Optimal solution found
Algorithm

1. Solve original problem
2. Is basis degenerate?
   - no: Update reduced problem.
   - yes: Decompose problem
3. Solve reduced problem
4. Update complementary problem
5. Solve complementary problem
6. Is s nonnegative?
   - yes: Optimal solution found
   - no: Nonnegative s implies no violated constraints.
Implemented within SoPlex
Simplex based LP solver developed at ZIB.

- Row representation available in SoPlex is necessary.
- Dual simplex used for original, reduced and complementary problems.
- No Phase 1 - degeneracy detected by subtracting bound shifts.

Complementary problem solved in the dual representation.

- Violated constraints are detected using a ratio test - comparing dual solutions from the reduced and complementary problems.
Outline

Introduction

Decomposition approach for the dual simplex

Solution approach and implementation

Results

Conclusion
Test instances

- Set of instances collected from numerous LP and MIP benchmark testsets - total of 1,226
- Only considered those solved in one hour using dual algorithm of SoPlex in row representation and by the decomposition algorithm.
- Removed instances solved by SoPlex in less than 0.05 seconds and displayed average degeneracy less than 20%.
- Final testset contained 157 instances.
Comparing the ratio of the number of iterations with respect to degeneracy - 157 instances
Conclusion

- Presented a decomposition approach designed to reduce the impact of degeneracy.
- Discussed the employed solution algorithm and the implementation in the freely available code of SoPlex.
- Demonstrated the potential of this approach to reduce the iterations of the dual simplex method.