Subexponential Lower Bounds for the Simplex Algorithm

Oliver Friedmann

Department of Computer Science,
Ludwig-Maximilians-Universität Munich, Germany.

January 20, 2011
Outline

1. LPs and Simplex algorithm
2. MDPs and Policy iteration
3. Random Edge
4. Random Facet and Least Entered
5. All is well that ends well?
LPs and Simplex algorithm
Outline

1. **LPs and Simplex algorithm**
   - Linear programming
   - Simplex algorithm
   - Pivoting rules
   - Technique

2. **MDPs and Policy iteration**

3. **Random Edge**

4. **Random Facet and Least Entered**

5. **All is well that ends well?**
Linear programming
Context

- Optimization of a linear objective subject to linear (in)equality constraints
- One of the most important computational problems
- **Simplex Algorithm** introduced by Dantzig in 1947 for solving LPs
- Weakly polytime algorithms: Ellipsoid Method by Khachiyan, Interior Point Method by Karmarkar
- No strongly polytime algorithm known
Linear programming

\[
\begin{align*}
\text{max} & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
\text{min} & \quad b^T y \\
\text{s.t.} & \quad A^T y \geq c \\
x \geq 0
\end{align*}
\]

- **Simplex** algorithm (Dantzig 1947)
- Ellipsoid algorithm (Khachiyan 1979)
- Interior-point algorithm (Karmakar 1984)
Simplex algorithm
Context

- Introduced by Dantzig in 1947 for solving LPs
- Fixpoint iteration algorithm
- Asymptotic complexity depends on the number of iterations
- Parameterized by pivoting rules
- Hirsch conjecture
Move up, along an edge, to a neighboring vertex, until reaching the top
Pivoting rules
Pivoting rules

Simplex method is parameterized by a pivoting rule.

Pivoting rule = method of choosing adjacent vertices with better objective

- only single-switching
- deterministic vs. randomized
- memorizing vs. oblivious
Deterministic rules

- Dantzig’s rule
- Bland’s rule
- Lexicographic rule
- many more...

Theorem (Klee-Minty (1972) et al.)

All known to require an exponential number of steps in the worst case.
Randomized rules

- **Random-Facet**: recursively find the optimum (see later) due to Kalai ('92), and Matoušek, Sharir and Welzl ('96)

- **Random-Edge**: choose a random improving edge

---

**Theorem (Friedmann-Hansen-Zwick (SODA’2011,...))**

There are (different) explicit LPs on which Random-Facet and Random-Edge require an expected subexponential number of iterations.
Memorizing rule

(taken from David Avis’ paper)

Theorem (Friedmann (IPCO’2011))

There are explicit LPs on which Least-Entered requires a subexponential number of iterations.
Technique
1. Lower bounds for Markov decision process policy iteration

2. Induce linear programs

3. Correspondence of simplex algorithm and policy iteration

originally: lower bounds for parity games first
MDPs and Policy iteration
Outline

1. LPs and Simplex algorithm
2. MDPs and Policy iteration
   - Markov decision processes
   - Policy iteration
   - MDPs as LPs
   - Summary
3. Random Edge
4. Random Facet and Least Entered
5. All is well that ends well?
Markov decision processes
Markov decision process

due to Shapley ’53, Bellman ’57, Howard ’60, ...

- **Controller** (circle) chooses outgoing action
- **Randomizer** (square) determines successive state
- **Objective**: maximize expected total reward

\[
\max \mathbb{E} \left[ \sum_{i=0}^{\infty} r(a_i) \right]
\]
Markov decision process

due to Shapley ’53, Bellman ’57, Howard ’60, ...

- Controller (circle) chooses outgoing action
- Randomizer (square) determines successive state
- Objective: maximize expected total reward

\[
\max \mathbb{E}\left[\sum_{i=0}^{\infty} r(a_i)\right]
\]

policy \( \sigma = \) choice of an action from each state
Policy iteration
Context

- Introduced by Howard in 1960 to solve Markov Decision Processes
- Transferred to many other areas by several authors
- Fixpoint iteration algorithm
- Asymptotic complexity depends on the number of iterations
- Parameterized by improvement rules
Policy valuations

\[ \text{val}_\sigma(a) = \frac{1}{2} b + \frac{1}{2} c = \]
\[ \text{val}_\sigma(b) = 2 + a = \]
\[ \text{val}_\sigma(c) = 6 + e = \]
\[ \text{val}_\sigma(d) = \frac{1}{2} b + \frac{1}{2} f = \]
\[ \text{val}_\sigma(e) = \frac{1}{2} c + \frac{1}{2} f = \]
\[ \text{val}_\sigma(f) = \]
Policy valuations

\[
\begin{align*}
\text{val}_\sigma(a) &= \frac{1}{2}b + \frac{1}{2}c = 14 \\
\text{val}_\sigma(b) &= 2 + a = 16 \\
\text{val}_\sigma(c) &= 6 + e = 12 \\
\text{val}_\sigma(d) &= \frac{1}{2}b + \frac{1}{2}f = 8 \\
\text{val}_\sigma(e) &= \frac{1}{2}c + \frac{1}{2}f = 6 \\
\text{val}_\sigma(f) &= 0
\end{align*}
\]
A \( \sigma \)-switch is an edge \( e \in E_0 \setminus \sigma \) not chosen by \( \sigma \).

Facts about switches

- **Comparability**: \( \text{val}\sigma \preceq \text{val}\sigma[e] \) or \( \text{val}\sigma[e] \preceq \text{val}\sigma \) for every \( \sigma \)-switch.
- **Easy check**: \( \text{val}\sigma \prec \text{val}\sigma[(v,w)] \) iff \( \text{val}\sigma(\sigma(v)) < \text{val}\sigma(w) \).

Improving switches: \( I(\sigma) = \{ e \mid \text{val}\sigma \prec \text{val}\sigma[e] \} \)

Theorem

- **Switching**: \( \emptyset \subsetneq J \subseteq I(\sigma) \) implies \( \text{val}\sigma \prec \text{val}\sigma[J] \).
- **Optimality**: \( I(\sigma) = \emptyset \) implies \( \sigma \) is optimal.

**Strategy improvement / Policy iteration**

1. **while** \( \sigma \) is not optimal **do**
2. \( \sigma \leftarrow \sigma[J] \) for some \( \emptyset \subset J \subseteq I(\sigma) \)
3. **end while**

**Complexity**: depends on the number of iterations.
Policy iteration

\[
\begin{align*}
\text{val}_\sigma(a) &= \frac{1}{2} b + \frac{1}{2} c = \\
\text{val}_\sigma(b) &= 2 + a = \\
\text{val}_\sigma(c) &= 6 + e = \\
\text{val}_\sigma(d) &= \frac{1}{2} b + \frac{1}{2} f = \\
\text{val}_\sigma(e) &= \frac{1}{2} c + \frac{1}{2} f = \\
\text{val}_\sigma(f) &= 
\end{align*}
\]
Policy iteration

\[ \text{val}_\sigma(a) = \frac{1}{2}b + \frac{1}{2}c = 14 \]
\[ \text{val}_\sigma(b) = 2 + a = 16 \]
\[ \text{val}_\sigma(c) = 6 + e = 12 \]
\[ \text{val}_\sigma(d) = \frac{1}{2}b + \frac{1}{2}f = 8 \]
\[ \text{val}_\sigma(e) = \frac{1}{2}c + \frac{1}{2}f = 6 \]
\[ \text{val}_\sigma(f) = 0 \]
Policy iteration

\[
\text{val}_\sigma(a) = \frac{1}{2} b + \frac{1}{2} c =
\]
\[
\text{val}_\sigma(b) = 2 + a =
\]
\[
\text{val}_\sigma(c) = 5 + d =
\]
\[
\text{val}_\sigma(d) = \frac{1}{2} b + \frac{1}{2} f =
\]
\[
\text{val}_\sigma(e) = \frac{1}{2} c + \frac{1}{2} f =
\]
\[
\text{val}_\sigma(f) =
\]
Policy iteration

\[ \text{val}_\sigma(a) = \frac{1}{2}b + \frac{1}{2}c = 16 \]
\[ \text{val}_\sigma(b) = 2 + a = 18 \]
\[ \text{val}_\sigma(c) = 5 + d = 12 \]
\[ \text{val}_\sigma(d) = \frac{1}{2}b + \frac{1}{2}f = 9 \]
\[ \text{val}_\sigma(e) = \frac{1}{2}c + \frac{1}{2}f = 7 \]
\[ \text{val}_\sigma(f) = 0 \]
Policy Iteration is parameterized by an improvement rule.

Improvement Rule = method of choosing improving switches

- Single-switching vs. Multi-switching
- Deterministic vs. Randomized
- Memorizing vs. Oblivious
Diameter

Question: theoretically possible to have polynomially many iterations?

Let $G$ be a game and $n$ be the number of nodes.

Definition: the diameter of $G$ is the least number of iterations required to solve $G$.

Diameter Theorem

The diameter of $G$ is less or equal to $n$. 
MDPs as LPs
Dual LP of unichain MDP

\[ \begin{align*}
(D) \quad & \min \sum_{i \in V} y_i \\
\text{s.t.} \quad & y_i - \sum_{j \in V} p_{j,a} y_j \geq r_a, \ i \in V, \ a \in A_i
\end{align*} \]

- \( V \) - set of all states
- \( A \) - set of all actions, \( A_i \) - set of actions from state \( i \)
- \( r_a \) - reward of action \( a \), \( p_{i,a} \) - transition probability from action \( a \) to state \( i \)
- \( y_i \) - value of state \( i \)
Primal LP of unichain MDP

\[
(P) \quad \begin{align*}
\max & \quad \sum_{a \in A} r_a x_a \\
\text{s.t.} & \quad \sum_{a \in A_i} x_a - \sum_{a \in A} p_{i,a} x_a = 1, \; i \in V \\
& \quad x_a \geq 0, \; a \in A
\end{align*}
\]

- \( x_a \) - expected number of times of taking action \( a \), when starting from all positions
Summary
Pivoting on the induced LP

Theorem

**Vertex** of bfs in the **primal LP** corresponds to **policy** in the **MDP**. **Improving switches** correspond to **pivoting steps**.

Theorem

**Optimal solution** of the **dual LP** corresponds to **optimal policies** in the **MDP**. **Policy** corresponds to **basic solution** of the dual.
## Policy iteration vs. Simplex method

<table>
<thead>
<tr>
<th></th>
<th>Policy iteration</th>
<th>Simplex method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pivoting Rules</td>
<td>single- and multi-switch</td>
<td>single-switch only</td>
</tr>
<tr>
<td>Diameter</td>
<td>linear</td>
<td>unknown / Hirsch conjecture</td>
</tr>
</tbody>
</table>
Random Edge

Random Edge
Outline

1. LPs and Simplex algorithm

2. MDPs and Policy iteration

3. Random Edge
   - Random edge rule
   - Motivation
   - Construction

4. Random Facet and Least Entered

5. All is well that ends well?
Random edge rule
Random edge rule

**RANDOM-EDGE rule**
Perform single switch **arbitrarily at random**.

**Context**
- Abstract lower bound: $2^\Omega(\sqrt[3]{n})$ (Matoušek, Szabó 2006)
Motivation
Construction principles

1. Process always reaches the sink (unichain MDPs)

2. Priority rewards on the nodes

\[ (-n)^5 \]
Cycle gadgets

\[ \text{val}_\sigma(b) \approx \text{val}_\sigma(d) \]
Cycle gadgets

\[ \text{val}_\sigma(b) = \text{val}_\sigma(a) \]
Binary Counting - How does it work?

\[ 101011 \]
Binary Counting - How does it work?

101011
↓
Setting
↓
101111
Binary Counting - How does it work?

101011
↓
Setting
↓
101111
↓
Resetting
↓
101100
Binary Counting - How does it work?

\[101011\]
\[\downarrow\]
Setting
\[101111\]
\[\downarrow\]
Resetting
\[101100\]
\[\downarrow\]
Activating
\[101100\]
Binary Counters

Binary Counting proceeds in three phases.

1. Set the least unset bit
2. Reset all lower (inactive) bits
3. Activate recently set bit

How to separate active from inactive bits? By access control.

- Attach access control structure to every bit
- Set bits are activated
- Recently set bit is not activated
- After resetting lower bits, recently set bit is activated
Construction
Random Edge Construction

Full construction

Diagram of a graph with nodes and edges labeled with variables such as $w$, $u$, $a$, $h$, $x$, $y$, $g$, $b$, $d$, $l$, and $s$. The diagram shows a series of connected nodes and arrows indicating the flow between them.
Cycle Gadget

\[ b_i \xrightarrow{1-\varepsilon/2} b_i \quad \text{and} \quad b_i \xleftarrow{\varepsilon} B_i \]

\[ B_i \xrightarrow{1-\varepsilon/2} b_i \]

\[ b_i, \ell_i \xrightarrow{\ell_i \varepsilon} b_i, 3 \xrightarrow{3\varepsilon} b_i, 2 \xrightarrow{2\varepsilon} b_{2,1} \xrightarrow{\varepsilon} b_i \]

\[ B_i \xleftarrow{\varepsilon} b_{2,1} \]

\[ b_i \xleftarrow{\varepsilon} b_i, \ell_i \]
Cycle Gadget

Cycles close one edge at a time
Cycle Gadget

Cycles close one edge at a time
Cycle Gadget

Cycles close one edge at a time
Cycle Gadget

Cycles close one edge at a time
Cycle Gadget

Cycles close one edge at a time
Cycle Gadget

Cycles close one edge at a time
Cycle Gadget

Cycles close one edge at a time
Cycle Gadget

Cycles close one edge at a time
Shorter cycles close faster
Cycle Gadget

\[
\begin{align*}
B_i & \quad \xrightarrow{\varepsilon} \quad b_i \\
\frac{1 - \varepsilon}{2} & \quad \xrightarrow{\varepsilon} \quad B_i
\end{align*}
\]

\[
\begin{align*}
b_i, \ell_i & \quad \xrightarrow{\ell_i \varepsilon} \quad b_i, 3 \\
3\varepsilon & \quad \xrightarrow{2\varepsilon} \quad b_i, 2 \\
2\varepsilon & \quad \xrightarrow{\varepsilon} \quad b_{2,1}
\end{align*}
\]

\[
\begin{align*}
\frac{1 - \varepsilon}{2} & \quad \xrightarrow{\varepsilon} \quad B_i
\end{align*}
\]
Cycle Gadget

Cycles open “simultaneously”
Cycle Gadget

Cycles open “simultaneously”
Cycle Gadget

Cycles open “simultaneously”
Cycle Gadget

Cycles open “simultaneously”
Cycle Gadget

Cycles open “simultaneously”
Increment phases: from 101011 to 101100

1 Setting: \[
\begin{align*}
B_k & \text{ counting cycle closes} \\
C_k & \text{ helper cycle closes}
\end{align*}
\]

2 Resetting: \[
\begin{align*}
U & \text{ lane realigns} \\
A_i & \text{ and } B_i \text{ cycles } (i < k) \text{ open}
\end{align*}
\]

3 Activating: \[
\begin{align*}
A_k & \text{ access cycle closes} \\
W & \text{ lane realigns} \\
C_i & \text{ cycles of unset bits open}
\end{align*}
\]
Cycles (opening and closing) and lanes compete with each other.

Supposed candidate has to win with high probability.

Solution: increase length of higher cycles, resulting in $O(n^4)$ vertices.

Work in progress: improved construction.
The number of improving step performed by \textsc{Random-Edge} on the MDPs (and LPs), which contain $O(n^4)$ vertices and edges, is $2^\Omega(n)$.

\textbf{Theorem (F.-Hansen-Zwick (2011))}
Random Facet and Least Entered
Outline

1. LPs and Simplex algorithm
2. MDPs and Policy iteration
3. Random Edge
4. Random Facet and Least Entered
   - Random Facet on LPs and MDPs
   - Random Facet construction
   - Least Entered
5. All is well that ends well?
Random Facet and Least Entered

Random Facet on LPs and MDPs

Oliver Friedmann (LMU)

Subexponential Lower Bounds for the

January 20, 2011
The algorithm
due to Kalai ('92), and Matoušek, Sharir and Welzl ('96)

**Random Facet and Least Entered**

**Random Facet on LPs and MDPs**

**The algorithm**

due to Kalai ('92), and Matoušek, Sharir and Welzl ('96)

**procedure** RANDOM-FACET\( (H,B) \)

\[
\begin{align*}
\text{if } H &= B \text{ then} \\
& \quad \text{return } B \\
\text{else} \\
& \quad \text{Choose } h \in H \setminus B \text{ at random} \\
& \quad B' \leftarrow \text{RANDOM-FACET}(H \setminus \{h\}, B) \\
& \quad \text{if } h \text{ is violated by } B' \text{ then} \\
& \quad \quad B'' \leftarrow \text{Basis}(B' \cup \{h\}) \\
& \quad \quad \text{return } \text{RANDOM-FACET}(H, B'') \\
& \quad \text{else} \\
& \quad \quad \text{return } B' \\
\end{align*}
\]

**Random Facet for LPs**

**procedure** RANDOM-FACET\( (G,\sigma) \)

\[
\begin{align*}
\text{if } E_0 &= \sigma \text{ then} \\
& \quad \text{return } \sigma \\
\text{else} \\
& \quad \text{Choose } e \in E_0 \setminus \sigma \text{ at random} \\
& \quad \sigma' \leftarrow \text{RANDOM-FACET}(G \setminus \{e\}, \sigma) \\
& \quad \text{if } \text{val}_{\sigma'} < \text{val}_{\sigma'}[e] \text{ then} \\
& \quad \quad \sigma'' \leftarrow \sigma'[e] \\
& \quad \quad \text{return } \text{RANDOM-FACET}(G, \sigma'') \\
& \quad \text{else} \\
& \quad \quad \text{return } \sigma' \\
\end{align*}
\]

**Random Facet for MDPs**
Context

Known results

- Upper bound: \(2^\mathcal{O}(n)\) (Kalai 1992)

- Abstract lower bound: \(2^{\Omega(\sqrt{n})}\) (Matoušek 1994)

Theorem (F.-Hansen-Zwick (2011))

**RANDOM-FACET** for MDPs and LPs is *subexponential*. 
Random Facet construction
Randomized counting

\[0|0|0|0|0\]
Randomized counting

-exclude bit 2

0|0|0|0|0

0|0|0|0|0
Randomized counting

0|0|0|0|0

exclude bit 2

0|0|0|0|0  1|1|1|1|1

count recursively
Randomized counting

- Exclude bit 2
- Set bit 2
- Count recursively

- $0|0|0|0|0$
- $0|0|1|0|0$
- $1|1|1|1|1$
- $1|1|1|0|0$

Analysis

- Counting $0$ equivalent to counting $0$
- Counting $1$ equivalent to counting $0$

Recurrence

$$f(n) = f(n-1) + 1$$

Complexity

$$f(n) \rightarrow e^{2 \cdot \sqrt{n} - \frac{1}{2}} \cdot \sqrt{\pi} \cdot n^{\frac{1}{4}}$$

for $n \rightarrow \infty$
Randomized counting

Random Facet and Least Entered

Random Facet construction

0|0|0|0|0

- exclude bit 2
- set bit 2

0|0|0|0|0 → 1|1|0|1|1
- count recursively

1|1|0|1|1
- count recursively

1|1|1|0|0

1|1|1|1|1

Complexity

$f(n) \rightarrow e^{2 \cdot \sqrt{n - 1}} / 2 \cdot \sqrt{\pi} \cdot n^{1/4}$ for $n \rightarrow \infty$
Randomized counting

- **0|0|0|0|0**
  - exclude bit 2
  - set bit 2
  - count recursively

Analysis

- Counting **0|0|0|0|0** equivalent to counting **0|0|0|0|0**
- Counting **1|1|1|1|1** equivalent to counting **0|0|0|0|0**
Randomized counting

Analysis

- Counting 0|0|0|0|0 equivalent to counting 0|0|0|0|0
Randomized counting

Analysis

- Counting 0|0|_|0|0 equivalent to counting 0|0|0|0
- Counting 1|1|1|0|0 equivalent to counting 0|0
Randomized counting

Analysis

- Counting 0|0|_|0|0 equivalent to counting 0|0|0|0
- Counting 1|1|1|0|0 equivalent to counting 0|0
- Recurrence \( f(n) = f(n - 1) + \frac{1}{n} \sum_{i=0}^{n-1} f(i) \)
Randomized counting

![Diagram showing randomized counting]

**Analysis**

- Counting 0|0|0|0|0 equivalent to counting 0|0|0|0|0
- Counting 1|1|1|0|0 equivalent to counting 0|0
- Recurrence \( f(n) = f(n - 1) + \frac{1}{n} \sum_{i=0}^{n-1} f(i) \)
- Complexity \( f(n) \rightarrow \frac{e^{2\cdot\sqrt{n}-\frac{1}{2}}}{2\cdot\sqrt{\pi}\cdot n^{\frac{3}{4}}} \) for \( n \rightarrow \infty \)
Simplified construction (for parity games)
Theorem (F.-Hansen-Zwick (2011))

The number of improving step performed by \textsc{Random-Facet} on the MDPs (and LPs) is $2^{\Omega(\sqrt{n}/\log(n))}$. 
Least Entered
Zadeh’s pivoting rule

Zadeh’s LEAST-ENTERED rule

Perform single switch that has been applied least often.
Fair counting

Problem:

- Flipping higher bits happens less often than flipping lower bits
- Zadeh’s rule switches higher bits before they are supposed to be switched

Solution:

- Represent every bit by two representatives
- Only one representative is actively working
- Inactive representative switches back and forth to catch up with the rest
- Both representatives change roles after flipping the represented bit
Full construction
Theorem (Friedmann (2011))

The number of improving step performed by Least-Entered on the MDPs, which contain $\mathcal{O}(n^2)$ vertices and edges, is $2^{\Omega(n)}$. 
All is well that ends well?
Concluding remarks

- **Game-theoretic** perspective helpful for the construction of lower bounds

- Lower bounds transfer to many other classes of determined games

- **RANDOM-EDGE** lower bound can be used as lower bound for **SWITCH-HALF**
All is well that ends well?

Relation to other games

- Turn-based Stochastic Games (TSG)
  - 2 players
- Discounted Payoff Games (DPG)
  - 2 players
- Mean Payoff Games (MPG)
  - 2 players
- Parity Games (PG)
  - 2 players
  - ∈ NP \cap coNP
- Deterministic Markov Decision Processes (DMDP)
  - 1 player
- Markov Decision Processes (MDP)
  - 1\frac{1}{2} players
- Linear Programming (LP)
- LP-type problems (LP\text{type})

\[ \in \text{NP} \cap \text{coNP} \]
\[ \in \text{P} \]
Open problems

- **Polytime** algorithm for two-player games and the like
- Strongly polytime algorithm for LPs (and MDPs)
- Resolving the **Hirsch conjecture**
- Find game-theoretic model with unresolved diameter bounds
All is well that ends well?

The slide usually called “the end”.

Thank you for listening!