# The simplex algorithm and the Hirsch conjecture: **Lecture 2**

Thomas Dueholm Hansen

MADALGO & CTIC Summer School

August 9, 2011

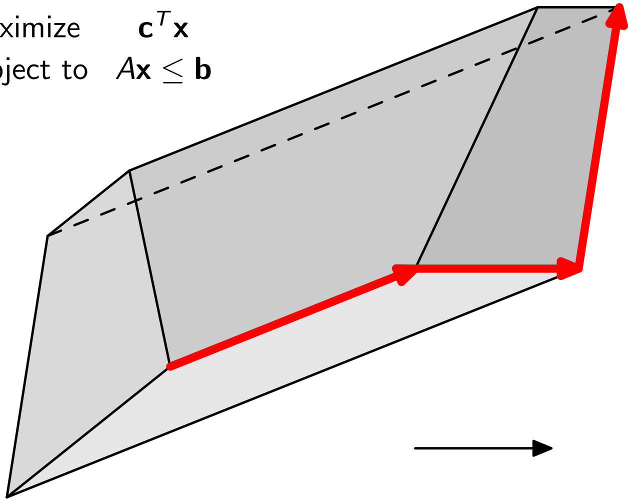
# Overview

- **Lecture 1:**
  - Introduction to linear programming and the simplex algorithm.
  - Pivoting rules.
  - The RandomFacet pivoting rule.
- **Lecture 2:**
  - The Hirsch conjecture.
  - Introduction to Markov decision processes (MDPs).
  - Upper bound for the LargestCoefficient pivoting rule for MDPs.
- **Lecture 3:**
  - Lower bounds for pivoting rules utilizing MDPs. Example: Bland's rule.
  - Lower bound for the RandomEdge pivoting rule.
  - Abstractions and related problems.

# The simplex algorithm, Dantzig (1947)

maximize $\mathbf{c}^T\mathbf{x}$

subject to $A\mathbf{x} \leq \mathbf{b}$



# The Hirsch conjecture

- The simplex algorithm motivates the study of the diameter of polytopes; bounds on the length of the best possible path to be picked by the simplex algorithm.

# The Hirsch conjecture

- The simplex algorithm motivates the study of the diameter of polytopes; bounds on the length of the best possible path to be picked by the simplex algorithm.
- The **distance** between two vertices $u$ and $v$ of a convex polytope $P$ is the fewest number of steps needed to get from $u$ to $v$ in the edge graph of $P$.
- The **diameter** of a convex polytope $P$ is the maximum distance between any two vertices $u$ and $v$.

# The Hirsch conjecture

- The simplex algorithm motivates the study of the diameter of polytopes; bounds on the length of the best possible path to be picked by the simplex algorithm.
- The **distance** between two vertices $u$ and $v$ of a convex polytope $P$ is the fewest number of steps needed to get from $u$ to $v$ in the edge graph of $P$.
- The **diameter** of a convex polytope $P$ is the maximum distance between any two vertices $u$ and $v$.
- Let $\Delta(d, n)$ and be the maximal diameter of any $d$-dimensional convex polytope defined by $n$ facets.

**Conjecture (Hirsch (1957))**

$\Delta(d, n) \leq n - d.$

- Klee and Walkup (1967) gave an example of an *unbounded* polytope with $d = 4$, $n = 8$ and diameter 5. In general they showed that $\Delta(d, n) \geq n - d + \lfloor d/5 \rfloor$ for $n \geq 2d$.

# The Hirsch conjecture

- Klee and Walkup (1967) gave an example of an *unbounded* polytope with $d = 4$, $n = 8$ and diameter 5. In general they showed that $\Delta(d, n) \geq n - d + \lfloor d/5 \rfloor$ for $n \geq 2d$.
- Todd (1980) transformed this result to bounded polytopes, but only for *nonincreasing* paths.

- Klee and Walkup (1967) gave an example of an *unbounded* polytope with $d = 4$, $n = 8$ and diameter 5. In general they showed that $\Delta(d, n) \geq n - d + \lfloor d/5 \rfloor$ for $n \geq 2d$.
- Todd (1980) transformed this result to bounded polytopes, but only for *nonincreasing* paths.
- Let $\Delta_b(d, n)$ and be the maximal diameter of any $d$-dimensional *bounded* convex polytope defined by $n$ facets.
- Bounded Hirsch conjecture:

**Conjecture**

$\Delta_b(d, n) \leq n - d$.

# The Hirsch conjecture

- Klee and Walkup (1967) gave an example of an *unbounded* polytope with $d = 4$, $n = 8$ and diameter 5. In general they showed that $\Delta(d,n) \geq n - d + \lfloor d/5 \rfloor$ for $n \geq 2d$.
- Todd (1980) transformed this result to bounded polytopes, but only for *nonincreasing* paths.
- Let $\Delta_b(d,n)$ and be the maximal diameter of any $d$-dimensional *bounded* convex polytope defined by $n$ facets.
- Bounded Hirsch conjecture:

**Conjecture**

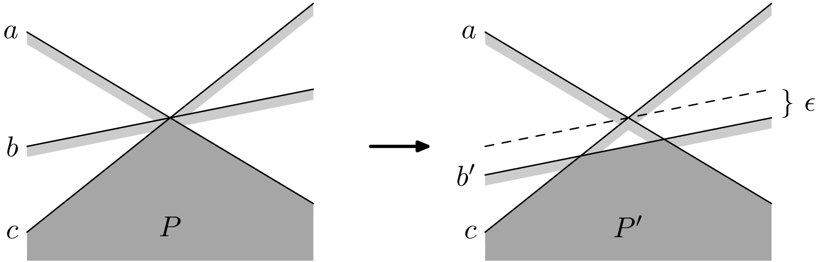
$\Delta_b(d,n) \leq n - d$.

- Polynomial Hirsch conjecture:

**Conjecture**

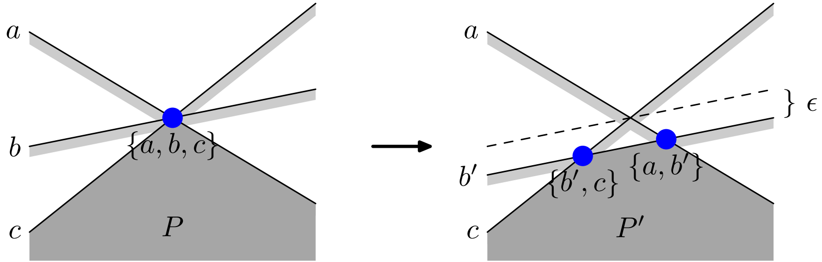
*There exists a polynomial $p$ such that $\Delta(d,n) \leq p(n)$.*

# Degeneracy



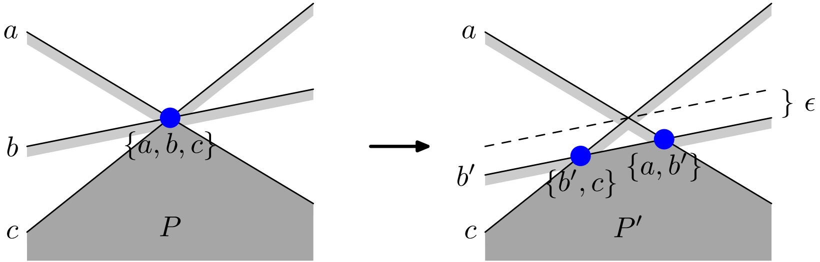

- Recall that a vertex is degenerate if it is contained in more than *d* facets.
- By slightly perturbing facets, degeneracy can be removed.

# Degeneracy




- Recall that a vertex is degenerate if it is contained in more than $d$ facets.
- By slightly perturbing facets, degeneracy can be removed.
- This operation corresponds to splitting degenerate vertices.
- If the pertubation is sufficiently small, the rest of the edge-graph of the polytope remains unchanged.

# Degeneracy




- Recall that a vertex is degenerate if it is contained in more than *d* facets.
- By slightly perturbing facets, degeneracy can be removed.
- This operation corresponds to splitting degenerate vertices.
- If the pertubation is sufficiently small, the rest of the edge-graph of the polytope remains unchanged.
- **Disclaimer:** I am generally not being formal about pertubations.

# Degeneracy

- Let $P'$ be obtained from $P$ by performing a pertubation:
  - **Splitting degenerate vertices:** Every vertex $v \in P$ maps to a set of vertices $S_v$ for $P'$.

# Degeneracy

- Let $P'$ be obtained from $P$ by performing a pertubation:
  - **Splitting degenerate vertices:** Every vertex $v \in P$ maps to a set of vertices $S_v$ for $P'$.
  - **Preserving the structure:** An edge $(u', v')$ for $P'$, with $u' \in S_u$ and $v' \in S_v$, corresponds to an edge $(u, v)$ for $P$ iff $u \neq v$.

# Degeneracy

- Let $P'$ be obtained from $P$ by performing a pertubation:
  - **Splitting degenerate vertices:** Every vertex $v \in P$ maps to a set of vertices $S_v$ for $P'$.
  - **Preserving the structure:** An edge $(u', v')$ for $P'$, with $u' \in S_u$ and $v' \in S_v$, corresponds to an edge $(u, v)$ for $P$ iff $u \neq v$.
- The distance between two vertices $u' \in S_u$ and $v' \in S_v$ in $P'$ is at least as large as the distance between the corresponding vertices $u$ and $v$ in $P$.

# Degeneracy

- Let $P'$ be obtained from $P$ by performing a pertubation:
  - **Splitting degenerate vertices:** Every vertex $v \in P$ maps to a set of vertices $S_v$ for $P'$.
  - **Preserving the structure:** An edge $(u', v')$ for $P'$, with $u' \in S_u$ and $v' \in S_v$, corresponds to an edge $(u, v)$ for $P$ iff $u \neq v$.
- The distance between two vertices $u' \in S_u$ and $v' \in S_v$ in $P'$ is at least as large as the distance between the corresponding vertices $u$ and $v$ in $P$.
- A polytope $P$ is **simple** if all its vertices are nondegenerate. I.e., every vertex is contained in exactly $d$ facets.

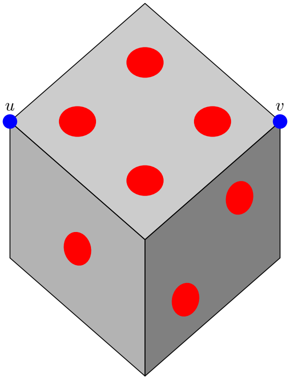
# Degeneracy

- Let $P'$ be obtained from $P$ by performing a pertubation:
  - **Splitting degenerate vertices:** Every vertex $v \in P$ maps to a set of vertices $S_v$ for $P'$.
  - **Preserving the structure:** An edge $(u', v')$ for $P'$, with $u' \in S_u$ and $v' \in S_v$, corresponds to an edge $(u, v)$ for $P$ iff $u \neq v$.
- The distance between two vertices $u' \in S_u$ and $v' \in S_v$ in $P'$ is at least as large as the distance between the corresponding vertices $u$ and $v$ in $P$.
- A polytope $P$ is **simple** if all its vertices are nondegenerate. I.e., every vertex is contained in exactly $d$ facets.
- Klee (1964): For every $d$-dimensional polytope $P$ with $n$ facets, there exists a simple $d$-polytope $P'$ with $n$ facets and diameter at least as large as the diameter of $P$.

# Degeneracy

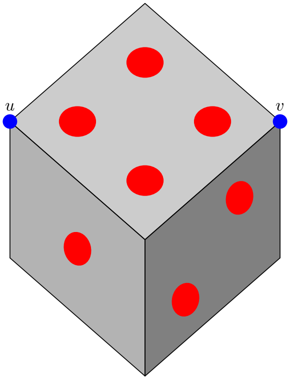
- Let $P'$ be obtained from $P$ by performing a pertubation:
  - **Splitting degenerate vertices:** Every vertex $v \in P$ maps to a set of vertices $S_v$ for $P'$.
  - **Preserving the structure:** An edge $(u', v')$ for $P'$, with $u' \in S_u$ and $v' \in S_v$, corresponds to an edge $(u, v)$ for $P$ iff $u \neq v$.
- The distance between two vertices $u' \in S_u$ and $v' \in S_v$ in $P'$ is at least as large as the distance between the corresponding vertices $u$ and $v$ in $P$.
- A polytope $P$ is **simple** if all its vertices are nondegenerate. I.e., every vertex is contained in exactly $d$ facets.
- Klee (1964): For every $d$-dimensional polytope $P$ with $n$ facets, there exists a simple $d$-polytope $P'$ with $n$ facets and diameter at least as large as the diameter of $P$.
  - Hence, when analyzing $\Delta(d, n)$ and $\Delta_b(d, n)$ we may restrict our attention to simple polytopes.

- For some $d$-polytope with $n$ facets, consider two vertices $u$ and $v$ that share $k$ facets.
- The distance between $u$ and $v$ is at most the length of the shortest path that stays within the $k$ shared facets, which is at most $\Delta(d-k, n-k)$.



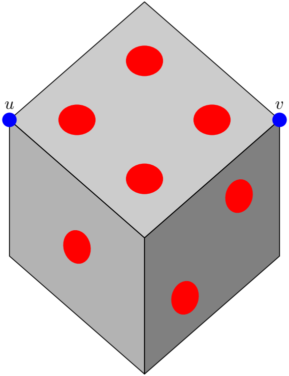
# Vertices sharing facets

- For some $d$-polytope with $n$ facets, consider two vertices $u$ and $v$ that share $k$ facets.
- The distance between $u$ and $v$ is at most the length of the shortest path that stays within the $k$ shared facets, which is at most $\Delta(d-k, n-k)$.
  - I.e., the $k$ facets define a $(d-k)$-face, which is itself a $(d-k)$ dimensional polytope.



# Vertices sharing facets

- For some $d$-polytope with $n$ facets, consider two vertices $u$ and $v$ that share $k$ facets.
- The distance between $u$ and $v$ is at most the length of the shortest path that stays within the $k$ shared facets, which is at most $\Delta(d-k, n-k)$.
  - I.e., the $k$ facets define a $(d-k)$-face, which is itself a $(d-k)$ dimensional polytope.

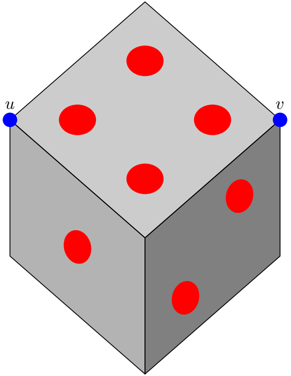



- Let $k = 2d - n > 0$, then:

$$\Delta(d, n) \leq \Delta(d-k, n-k) = \Delta(n-d, 2(n-d))$$

# Vertices sharing facets

- For some $d$-polytope with $n$ facets, consider two vertices $u$ and $v$ that share $k$ facets.
- The distance between $u$ and $v$ is at most the length of the shortest path that stays within the $k$ shared facets, which is at most $\Delta(d-k, n-k)$.
  - I.e., the $k$ facets define a $(d-k)$-face, which is itself a $(d-k)$ dimensional polytope.
- Let $k = 2d - n > 0$, then:

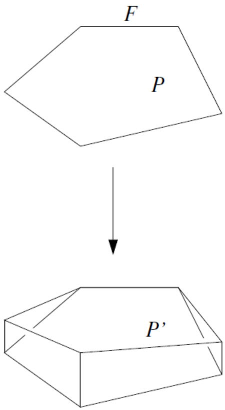



$$\Delta(d,n) \leq \Delta(d-k, n-k) = \Delta(n-d, 2(n-d))$$

$$\Delta_b(d,n) \leq \Delta_b(d-k, n-k) = \Delta_b(n-d, 2(n-d))$$

# The wedge operation

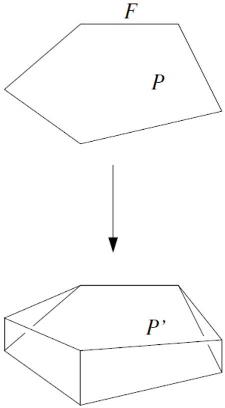
- Klee and Walkup (1967) defined a *wedge* operation for creating polytopes[a]:
  - Let $P$ be a bounded $d$-polytope with $n$ facets, and let $F$ be a facet of $P$.
  - A new polytope $P'$ in dimension $d + 1$ is created by copying vertices not in $F$ and "lifting" the copies to a new hyperplane.




[a] The picture is from a presentation by F. Santos: `http://tinyurl.com/3uk9grc`

# The wedge operation

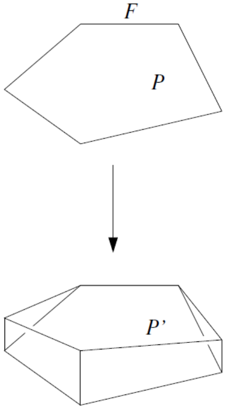
- Klee and Walkup (1967) defined a *wedge* operation for creating polytopes[a]:
  - Let $P$ be a bounded $d$-polytope with $n$ facets, and let $F$ be a facet of $P$.
  - A new polytope $P'$ in dimension $d+1$ is created by copying vertices not in $F$ and "lifting" the copies to a new hyperplane.
  - I.e., replace $a_F^T x \leq b_F$ by the two constraints $x_{d+1} \geq 0$ and $a_F^T x + x_{d+1} \leq b_F$, where $x \in \mathbb{R}^d$ and $x_{d+1}$ is a new variable.




[a] The picture is from a presentation by F. Santos: `http://tinyurl.com/3uk9grc`

# The wedge operation

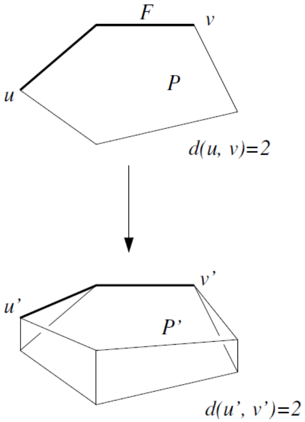
- Klee and Walkup (1967) defined a *wedge* operation for creating polytopes[a]:
  - Let $P$ be a bounded $d$-polytope with $n$ facets, and let $F$ be a facet of $P$.
  - A new polytope $P'$ in dimension $d+1$ is created by copying vertices not in $F$ and "lifting" the copies to a new hyperplane.
  - I.e., replace $a_F^T x \leq b_F$ by the two constraints $x_{d+1} \geq 0$ and $a_F^T x + x_{d+1} \leq b_F$, where $x \in \mathbb{R}^d$ and $x_{d+1}$ is a new variable.
  - $P'$ has $n+1$ facets.




[a] The picture is from a presentation by F. Santos: `http://tinyurl.com/3uk9grc`

# The wedge operation

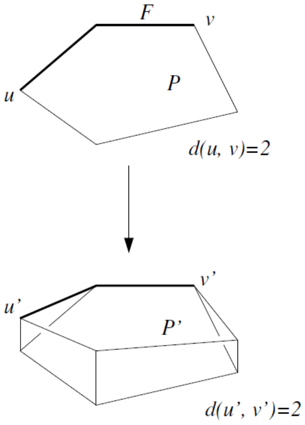
- Let $P'$ be obtained from $P$ by performing a wedge operation[a], and let $u'$ and $v'$ be any two vertices of $P'$.
- The distance between $u'$ and $v'$ is at least as large as the distance between the corresponding vertices $u$ and $v$ in $P$.




[a] The picture is from a presentation by F. Santos: `http://tinyurl.com/3uk9grc`

# The wedge operation

- Let $P'$ be obtained from $P$ by performing a wedge operation[a], and let $u'$ and $v'$ be any two vertices of $P'$.
- The distance between $u'$ and $v'$ is at least as large as the distance between the corresponding vertices $u$ and $v$ in $P$.
- Hence, $\Delta_b(d,n) \leq \Delta_b(d+1,n+1)$.




[a]The picture is from a presentation by F. Santos: `http://tinyurl.com/3uk9grc`

# The $d$-step conjecture

- If $k = n - 2d > 0$, then repeated use of the wedge operation gives:

$$\Delta_b(d, n) \leq \Delta_b(d + k, n + k) = \Delta_b(n - d, 2(n - d))$$

# The $d$-step conjecture

- If $k = n - 2d > 0$, then repeated use of the wedge operation gives:

$$\Delta_b(d, n) \leq \Delta_b(d + k, n + k) = \Delta_b(n - d, 2(n - d))$$

- When combined with the bound obtained when $n < 2d$, we get for all $d$ and $n$:

$$\Delta_b(d, n) \leq \Delta_b(n - d, 2(n - d))$$

# The $d$-step conjecture

- If $k = n - 2d > 0$, then repeated use of the wedge operation gives:

$$\Delta_b(d,n) \le \Delta_b(d+k, n+k) = \Delta_b(n-d, 2(n-d))$$

- When combined with the bound obtained when $n < 2d$, we get for all $d$ and $n$:

$$\Delta_b(d,n) \le \Delta_b(n-d, 2(n-d))$$

**Theorem (Klee and Walkup (1967))**

*The bounded Hirsch conjecture can be equivalently stated as $\Delta_b(d, 2d) \le d$, for all $d$.*

# The Hirsch conjecture

- Klee (1965): $\Delta_b(d, n) \leq n - d$ for $d \leq 3$.
- Klee and Walkup (1967): $\Delta_b(d, n) \leq n - d$ for $n - d \leq 5$.

# The Hirsch conjecture

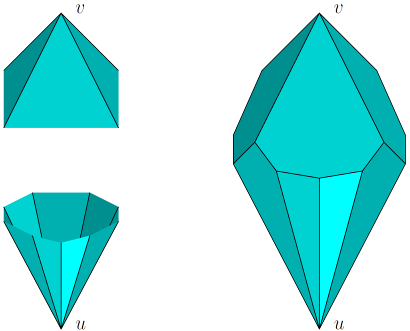
- Klee (1965): $\Delta_b(d,n) \le n-d$ for $d \le 3$.
- Klee and Walkup (1967): $\Delta_b(d,n) \le n-d$ for $n-d \le 5$.
  - Bremner and Schewe (2008): $\Delta_b(d,n) \le n-d$ for $n-d \le 6$.

# The Hirsch conjecture

- Klee (1965): $\Delta_b(d,n) \le n-d$ for $d \le 3$.
- Klee and Walkup (1967): $\Delta_b(d,n) \le n-d$ for $n-d \le 5$.
  - Bremner and Schewe (2008): $\Delta_b(d,n) \le n-d$ for $n-d \le 6$.
- Santos (2010) gave an example of a bounded polytope with $d = 43$, $n = 86$, and diameter at least 44. In general, Santos shows that for fixed $d$ and $\epsilon$, $\Delta_b(d,n) \ge (1+\epsilon)(n-d)$.

- Klee (1965): $\Delta_b(d,n) \leq n-d$ for $d \leq 3$.
- Klee and Walkup (1967): $\Delta_b(d,n) \leq n-d$ for $n-d \leq 5$.
  - Bremner and Schewe (2008): $\Delta_b(d,n) \leq n-d$ for $n-d \leq 6$.
- Santos (2010) gave an example of a bounded polytope with $d=43$, $n=86$, and diameter at least 44. In general, Santos shows that for fixed $d$ and $\epsilon$, $\Delta_b(d,n) \geq (1+\epsilon)(n-d)$.
  - Matschke, Santos and Weibel (2011): An example with $d=20$, $n=40$, 36442 vertices, and diameter 21. This gives $\epsilon \approx 1/20$.

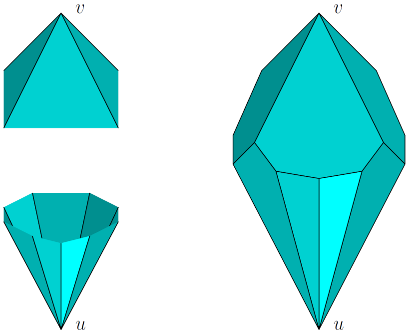
# Spindles




- A $d$-polytope with $n \geq 2d$ facets is called a **spindle** if it has two vertices $u$ and $v$, such that $u$ and $v$ do not share a facet, and all facets of $P$ contain either $u$ or $v$.[1]

[1] The picture is from Santos (2010).

# Spindles




- A $d$-polytope with $n \geq 2d$ facets is called a **spindle** if it has two vertices $u$ and $v$, such that $u$ and $v$ do not share a facet, and all facets of $P$ contain either $u$ or $v$.[1]
- The **length** of a spindle is the distance from $u$ to $v$.

[1] The picture is from Santos (2010).

# The counterexample, Santos (2010)

**Theorem (The "Santos-wedge")**

*If there exists a spindle of dimension $d$, with $n > 2d$ facets, and length $\ell$, then there exists a spindle of dimension $d+1$, with $n+1$ facets, and length at least $\ell + 1$.*

# The counterexample, Santos (2010)

**Theorem (The "Santos-wedge")**

*If there exists a spindle of dimension $d$, with $n > 2d$ facets, and length $\ell$, then there exists a spindle of dimension $d+1$, with $n+1$ facets, and length at least $\ell + 1$.*

- Using the theorem repeatedly $n - 2d$ times gives: If there exists a spindle with parameters $d, n, \ell$, then there exists a $(n-d)$-dimensional spindle with $2(n-d)$ facets and length $\ell + n - 2d$.

# The counterexample, Santos (2010)

**Theorem (The "Santos-wedge")**

*If there exists a spindle of dimension $d$, with $n > 2d$ facets, and length $\ell$, then there exists a spindle of dimension $d+1$, with $n+1$ facets, and length at least $\ell+1$.*

- Using the theorem repeatedly $n-2d$ times gives: If there exists a spindle with parameters $d, n, \ell$, then there exists a $(n-d)$-dimensional spindle with $2(n-d)$ facets and length $\ell + n - 2d$.
- Hence, if there exists a $d$-dimensional spindle of length $\ell > d$, then the bounded Hirsch conjecture is false.

# The counterexample, Santos (2010)

**Theorem (The "Santos-wedge")**

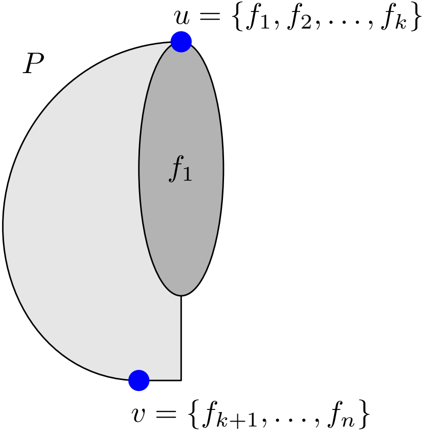
*If there exists a spindle of dimension $d$, with $n > 2d$ facets, and length $\ell$, then there exists a spindle of dimension $d+1$, with $n+1$ facets, and length at least $\ell + 1$.*

- Using the theorem repeatedly $n - 2d$ times gives: If there exists a spindle with parameters $d, n, \ell$, then there exists a $(n-d)$-dimensional spindle with $2(n-d)$ facets and length $\ell + n - 2d$.
- Hence, if there exists a $d$-dimensional spindle of length $\ell > d$, then the bounded Hirsch conjecture is false.

**Theorem**

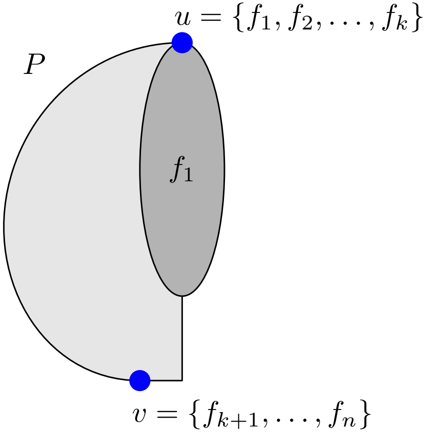
*There exists a 5-dimensional spindle with 48 facets and length 6.*

# The "Santos-wedge"



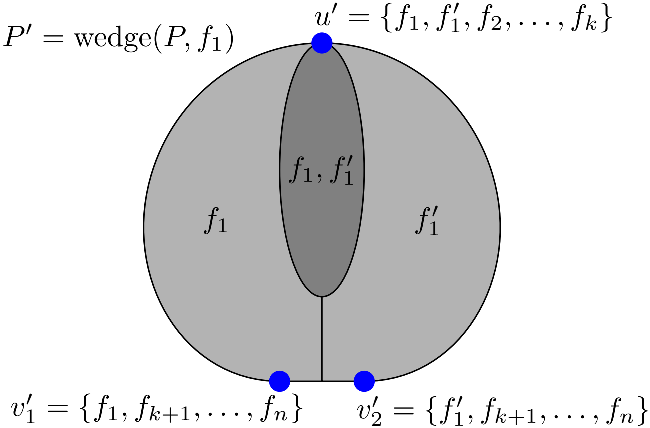

- Let $P$ be a $d$-dimensional spindle with $n > 2d$ facets and length $\ell$.

# The "Santos-wedge"



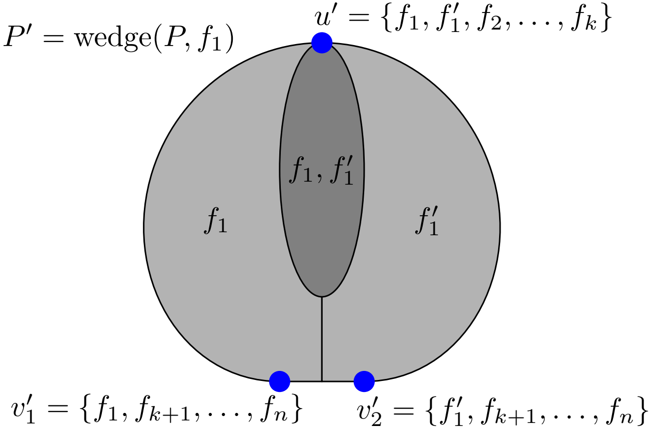

- Let $P$ be a $d$-dimensional spindle with $n > 2d$ facets and length $\ell$.
- Then at least one of the two antipodal vertices $u$ and $v$ is degenerate. Assume $v$ is degenerate.

# The “Santos-wedge”

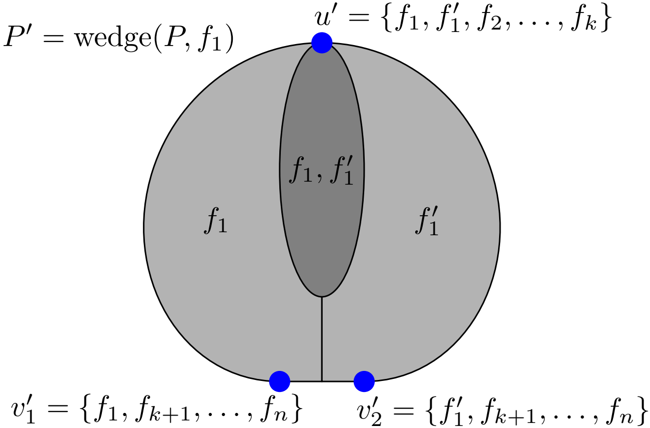



- Construct a polytope $P'$ as the wedge of $P$ and one of the facets $f_1$ containing $u$.

# The "Santos-wedge"

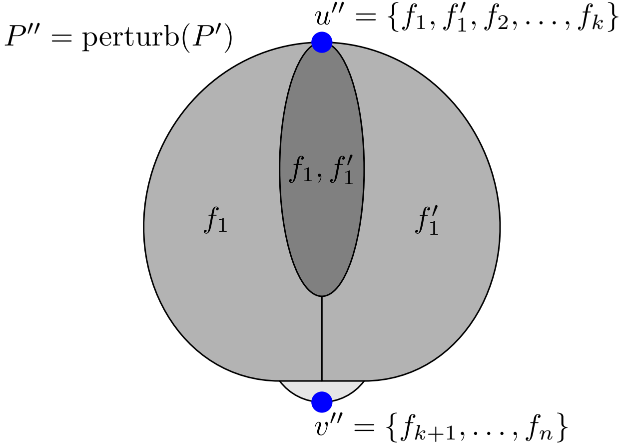



- Construct a polytope $P'$ as the wedge of $P$ and one of the facets $f_1$ containing $u$.
- $P'$ is not a spindle since there are two vertices $v_1'$ and $v_2'$ corresponding to $v$, both sharing a facet with $u'$.



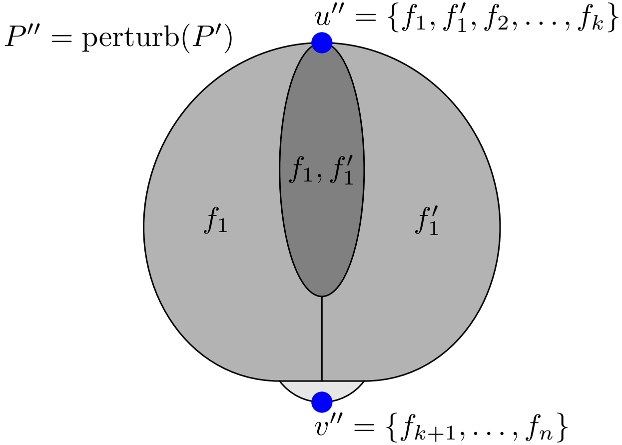

- Since $v$ was degenerate, $v_1'$ and $v_2'$ are also degenerate.

# The "Santos-wedge"



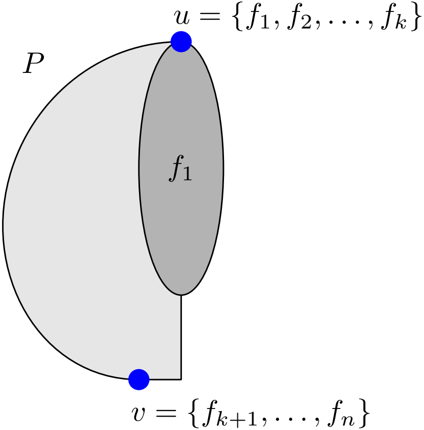

- Since $v$ was degenerate, $v'_1$ and $v'_2$ are also degenerate.
- Perturb a facet $f_i$, $i > k$, such that the only degenerate vertices in $f_i$ are $v'_1$ and $v'_2$. If no such facet is readily available, a preceding pertubation is made.

# The "Santos-wedge"



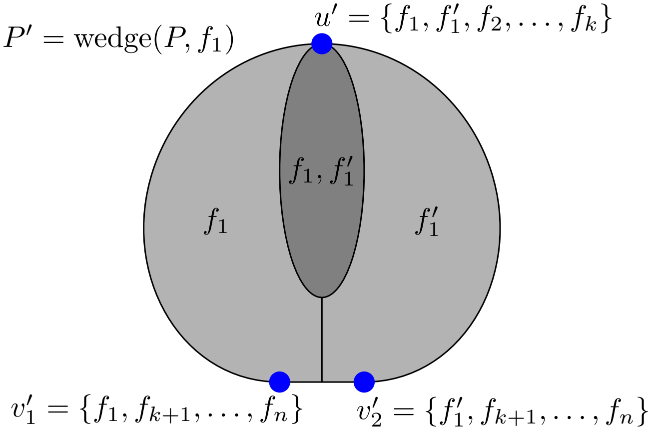

- The pertubation creates a vertex $v'' = \{f_{k+1}, \ldots, f_n\}$, and the resulting polytope $P''$ is, thus, a $(d+1)$-dimensional spindle with $n+1$ facets.
- Claim: The length of $P''$ is at least $\ell + 1$.

# The "Santos-wedge"



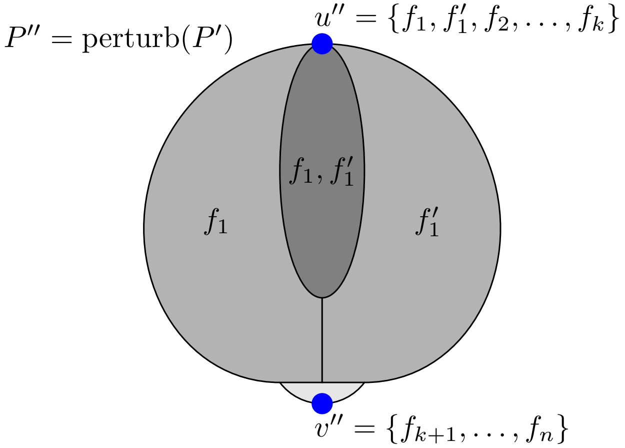

- All vertices of $P$ different from $v$ shared a facet with $u$.

# The "Santos-wedge"



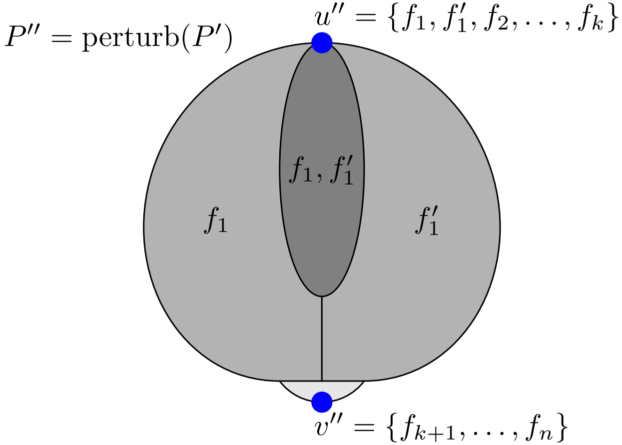

- All vertices of $P$ different from $v$ shared a facet with $u$.
- Therefore, all vertices of $P'$ different from $v_1'$ and $v_2'$ contain one of the facets $f_1$ or $f_1'$ and one more facet shared with $u'$.

# The "Santos-wedge"



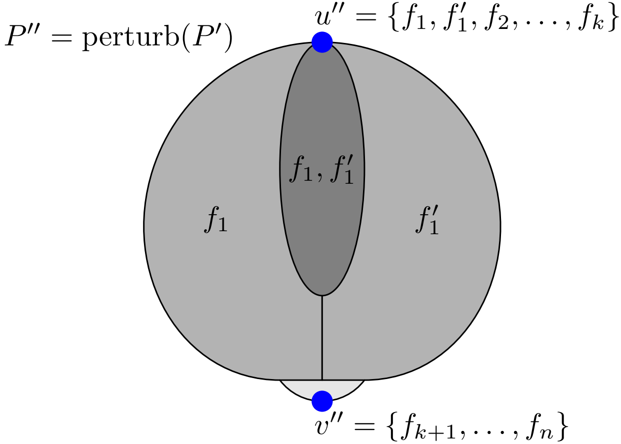

- All vertices of $P$ different from $v$ shared a facet with $u$.
- Therefore, all vertices of $P'$ different from $v_1'$ and $v_2'$ contain one of the facets $f_1$ or $f_1'$ and one more facet shared with $u'$.
- $v''$ is at distance at least 2 from all such vertices.

# The "Santos-wedge"




- Since only $v_1'$ and $v_2'$ were split during the (latest) pertubation, all neighbours of $v''$ also originated from either $v_1'$ or $v_2'$.

# The "Santos-wedge"




- Since only $v'_1$ and $v'_2$ were split during the (latest) pertubation, all neighbours of $v''$ also originated from either $v'_1$ or $v'_2$.
- Hence, an additional first step from $v''$ has been added, and the length of $P''$ has been increased compared to $P$.

# Upper bounding the diameter of polytopes

- The RandomFacet pivoting rule gives a subexponential upper bound on the diameter of polytopes, $\Delta(d,n) \leq 2^{O(\sqrt{(n-d)\log n})}$.

# Upper bounding the diameter of polytopes

- The RandomFacet pivoting rule gives a subexponential upper bound on the diameter of polytopes, $\Delta(d, n) \leq 2^{O(\sqrt{(n-d)\log n})}$.
- Kalai and Kleitman (1992) showed a *quasi-polynomial* upper bound, $\Delta(d, n) \leq n^{\log d + 1}$.

# Upper bounding the diameter of polytopes

- The RandomFacet pivoting rule gives a subexponential upper bound on the diameter of polytopes, $\Delta(d,n) \le 2^{O(\sqrt{(n-d)\log n})}$.
- Kalai and Kleitman (1992) showed a *quasi-polynomial* upper bound, $\Delta(d,n) \le n^{\log d + 1}$.
- Larman (1970): $\Delta(d,n) \le 2^{d-3} n$
- Barnette (1974): $\Delta(d,n) \le \frac{2^{d-2}}{3} n$

# Upper bounding the diameter of polytopes

- The RandomFacet pivoting rule gives a subexponential upper bound on the diameter of polytopes, $\Delta(d,n) \leq 2^{O(\sqrt{(n-d)\log n})}$.
- Kalai and Kleitman (1992) showed a *quasi-polynomial* upper bound, $\Delta(d,n) \leq n^{\log d + 1}$.
- Larman (1970): $\Delta(d,n) \leq 2^{d-3} n$
- Barnette (1974): $\Delta(d,n) \leq \frac{2^{d-2}}{3} n$
- The polynomial Hirsch conjecture has yet to be resolved. It is the subject of the **polymath3** project:

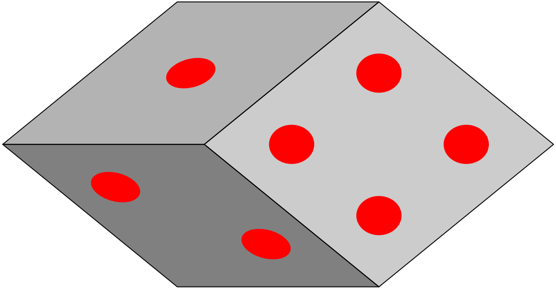
  `gilkalai.wordpress.com/category/polymath3/`

# Upper bounding the diameter of polytopes

- The RandomFacet pivoting rule gives a subexponential upper bound on the diameter of polytopes, $\Delta(d,n) \leq 2^{O(\sqrt{(n-d)\log n})}$.
- Kalai and Kleitman (1992) showed a *quasi-polynomial* upper bound, $\Delta(d,n) \leq n^{\log d+1}$.
- Larman (1970): $\Delta(d,n) \leq 2^{d-3}n$
- Barnette (1974): $\Delta(d,n) \leq \frac{2^{d-2}}{3}n$
- The polynomial Hirsch conjecture has yet to be resolved. It is the subject of the **polymath3** project:

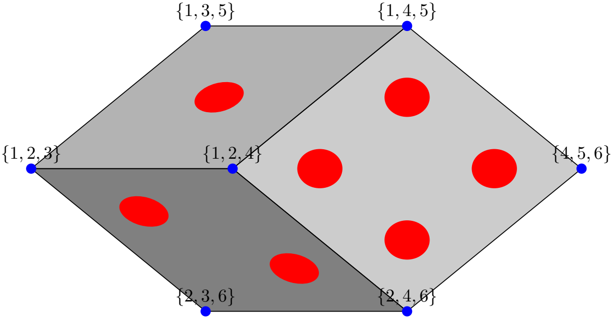
  gilkalai.wordpress.com/category/polymath3/
- We next prove the bounds of Kalai and Kleitman (1992) and Larman (1970) in an abstract framework by Eisenbrand, Hähnle, Razborov and Rothvoß (2009).

# Connected layer families



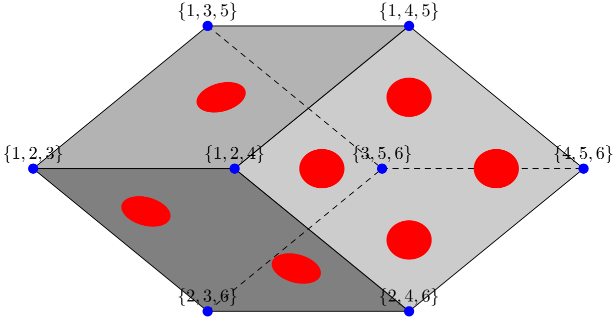
- Consider a simple polytope with $n$ facets, indexed from 1 to $n$, in $d$-space.

# Connected layer families



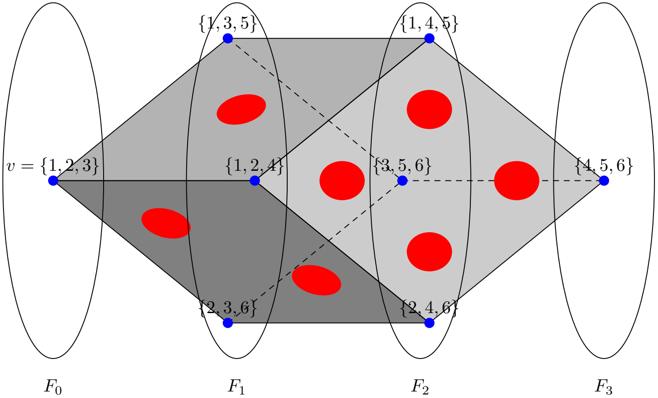

- Consider a simple polytope with $n$ facets, indexed from 1 to $n$, in $d$-space.
- Vertices are identified by subsets $v \subseteq \{1, \ldots, n\}$ of size $d$.

# Connected layer families



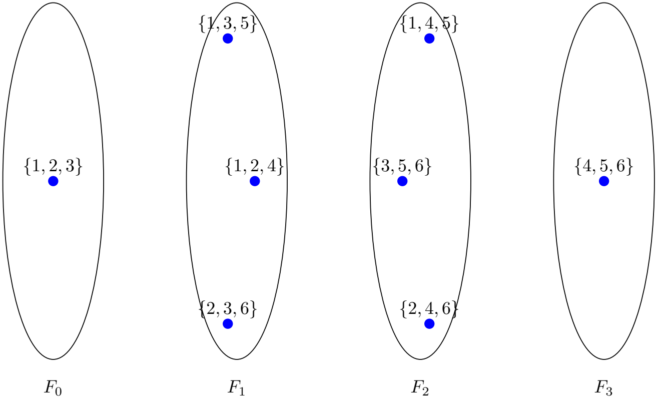

- Consider a simple polytope with $n$ facets, indexed from 1 to $n$, in $d$-space.
- Vertices are identified by subsets $v \subseteq \{1, \ldots, n\}$ of size $d$.

# Connected layer families



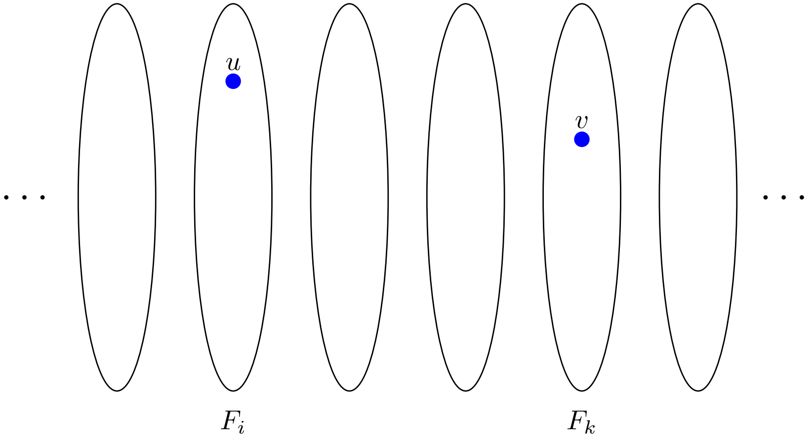

- Pick some vertex $v$, and let $F_i$ be the set of vertices at distance $i$ from $v$.

# Connected layer families



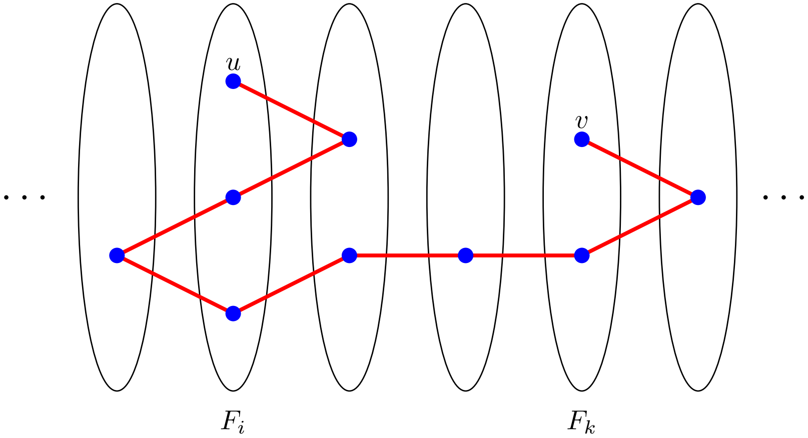

- Pick some vertex $v$, and let $F_i$ be the set of vertices at distance $i$ from $v$.
- $F_i$ is a family of subsets of $\{1, \ldots, n\}$ of size $d$.

# Connected layer families



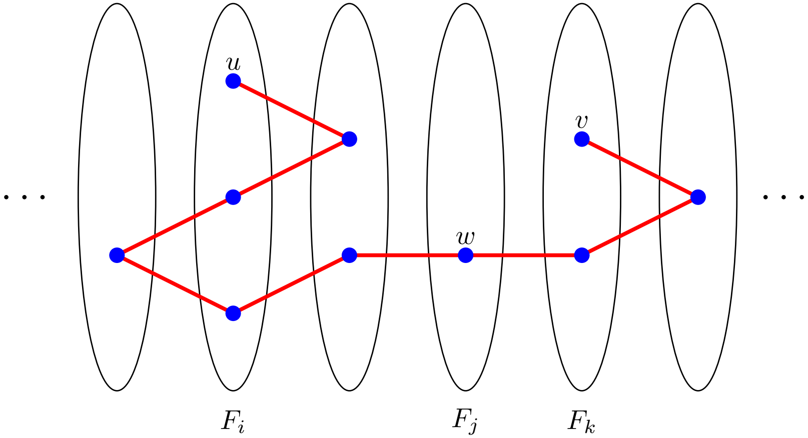

- Consider two vertices $u$ and $v$ in different families $F_i$ and $F_k$.

# Connected layer families




- Suppose *u* and *v* share *k* facets. Then there is a path from *u* to *v* in the polytope that stays within these *k* facets. The path cannot skip a layer.

# Connected layer families




$$\forall i < j < k \;\; \forall u \in F_i, v \in F_k \;\; \exists w \in F_j : \quad u \cap v \subseteq w$$

# Connected layer families

- A $d$-dimensional **connected layer family** (CLF) $\mathcal{F}$ with $n$ symbols and **height** $t$ is defined as:
  - $t$ disjoint, nonempty families, $F_1, \ldots, F_t$, of subsets of $\{1, 2, \ldots, n\}$ of size $d$ satisfying the connectivity restriction:

$$\forall i < j < k \;\; \forall u \in F_i, v \in F_k \;\; \exists w \in F_j : \quad u \cap v \subseteq w$$

# Connected layer families

- A $d$-dimensional **connected layer family** (CLF) $\mathcal{F}$ with $n$ symbols and **height** $t$ is defined as:
  - $t$ disjoint, nonempty families, $F_1, \ldots, F_t$, of subsets of $\{1, 2, \ldots, n\}$ of size $d$ satisfying the connectivity restriction:

$$\forall i < j < k \;\; \forall u \in F_i, v \in F_k \;\; \exists w \in F_j : \quad u \cap v \subseteq w$$

- Let $\Delta_{clf}(d, n)$ be the maximum integer $t$ such that there exists a $d$-dimensional CLF with $n$ symbols and height $t$.

# Connected layer families

- A $d$-dimensional **connected layer family** (CLF) $\mathcal{F}$ with $n$ symbols and **height** $t$ is defined as:
  - $t$ disjoint, nonempty families, $F_1, \ldots, F_t$, of subsets of $\{1, 2, \ldots, n\}$ of size $d$ satisfying the connectivity restriction:

$$\forall i < j < k \;\; \forall u \in F_i, v \in F_k \;\; \exists w \in F_j : \quad u \cap v \subseteq w$$

- Let $\Delta_{clf}(d, n)$ be the maximum integer $t$ such that there exists a $d$-dimensional CLF with $n$ symbols and height $t$.
- Then $\Delta(d, n) \leq \Delta_{clf}(d, n) + 1$.

# Induced connected layer families

- We say that a symbol $s \in \{1, \ldots, n\}$ is **active** in layer $i$ if there exists $v \in F_i$ with $s \in v$.

# Induced connected layer families

- We say that a symbol $s \in \{1, \dots, n\}$ is **active** in layer $i$ if there exists $v \in F_i$ with $s \in v$.
- Let $L(s)$ and $U(s)$ be the lowest and highest index for which $s$ is active.

# Induced connected layer families

- We say that a symbol $s \in \{1, \ldots, n\}$ is **active** in layer $i$ if there exists $v \in F_i$ with $s \in v$.
- Let $L(s)$ and $U(s)$ be the lowest and highest index for which $s$ is active.
- The **induced** CLF $\mathcal{F}^s$ (or facet) for a symbol $s \in \{1, \ldots, n\}$ is obtained by throwing away all subsets not containing $s$. We are left with:

$$F_i^s = \{v \in F_i \mid s \in v\}, \quad \text{for } L(s) \leq i \leq U(s)$$

# Induced connected layer families

- We say that a symbol $s \in \{1, \ldots, n\}$ is **active** in layer $i$ if there exists $v \in F_i$ with $s \in v$.
- Let $L(s)$ and $U(s)$ be the lowest and highest index for which $s$ is active.
- The **induced** CLF $\mathcal{F}^s$ (or facet) for a symbol $s \in \{1, \ldots, n\}$ is obtained by throwing away all subsets not containing $s$. We are left with:

$$F_i^s = \{v \in F_i \mid s \in v\}, \quad \text{for } L(s) \leq i \leq U(s)$$

- $\mathcal{F}^s$ satisfies the connectivity restriction: Every subset $w$ used to satisfy the connectivity restriction for $\mathcal{F}$ for $u \in F_i^s$ and $v \in F_k^s$ must contain $s$ and is, thus, included in $\mathcal{F}^s$.

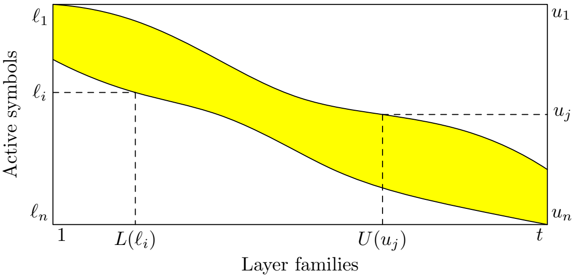
# Induced connected layer families

- We say that a symbol $s \in \{1, \ldots, n\}$ is **active** in layer $i$ if there exists $v \in F_i$ with $s \in v$.
- Let $L(s)$ and $U(s)$ be the lowest and highest index for which $s$ is active.
- The **induced** CLF $\mathcal{F}^s$ (or facet) for a symbol $s \in \{1, \ldots, n\}$ is obtained by throwing away all subsets not containing $s$. We are left with:

$$F_i^s = \{v \in F_i \mid s \in v\}, \quad \text{for } L(s) \leq i \leq U(s)$$

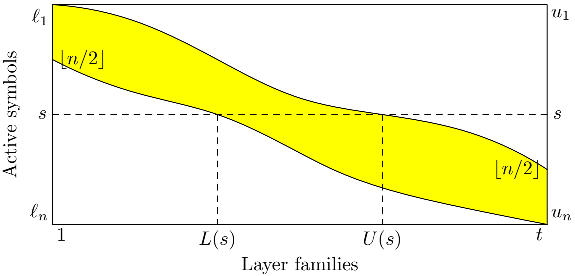
- $\mathcal{F}^s$ satisfies the connectivity restriction: Every subset $w$ used to satisfy the connectivity restriction for $\mathcal{F}$ for $u \in F_i^s$ and $v \in F_k^s$ must contain $s$ and is, thus, included in $\mathcal{F}^s$.
- By removing $s$ from all sets of $\mathcal{F}^s$, we get a $d-1$ dimensional connected layer family with $n-1$ symbols and height $U(s) - L(s) + 1 \leq \Delta_{clf}(d-1, n-1)$.

# Quasi-polynomial upper bound



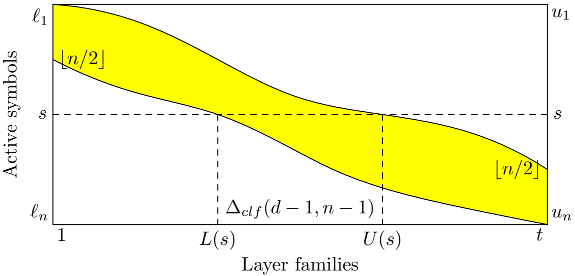

- Let $\mathcal{L} = \ell_1, \ell_2, \ldots, \ell_n$ and $\mathcal{U} = u_1, u_2, \ldots, u_n$ be the lists of symbols sorted in increasing order according to $L(s)$ and $U(s)$, respectively.

# Quasi-polynomial upper bound



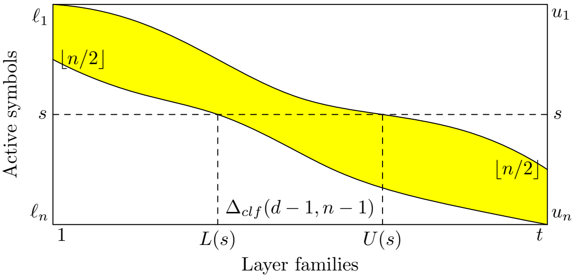

- Let $\mathcal{L} = \ell_1, \ell_2, \ldots, \ell_n$ and $\mathcal{U} = u_1, u_2, \ldots, u_n$ be the lists of symbols sorted in increasing order according to $L(s)$ and $U(s)$, respectively.
- By the pigeonhole principle there exists a common symbol $s$ among the first $\lfloor n/2 \rfloor + 1$ symbols of $\mathcal{L}$ and the last $\lfloor n/2 \rfloor + 1$ symbols of $\mathcal{U}$.

# Quasi-polynomial upper bound



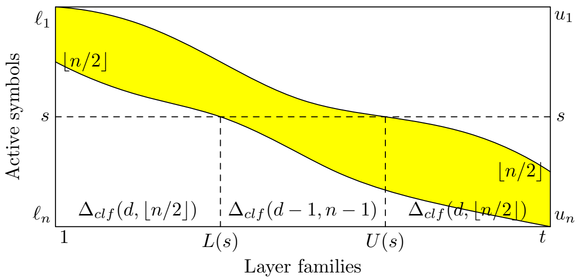

- The length of the interval from $L(s)$ to $U(s)$ is the height of $\mathcal{F}^s$ which is at most $\Delta_{clf}(d-1, n-1)$.

# Quasi-polynomial upper bound



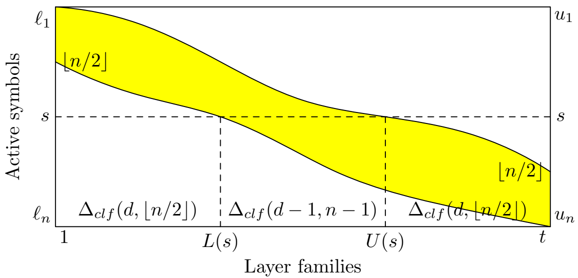

- The length of the interval from $L(s)$ to $U(s)$ is the height of $\mathcal{F}^s$ which is at most $\Delta_{clf}(d-1, n-1)$.
- Before $L(s)$ and after $U(s)$ there are at most $\lfloor n/2 \rfloor$ active symbols.

# Quasi-polynomial upper bound



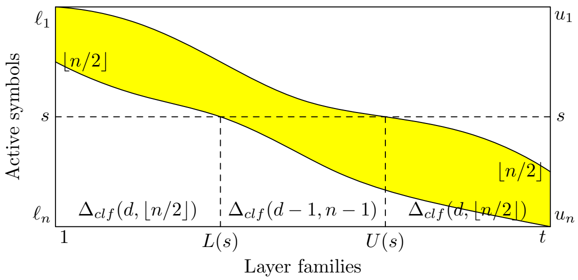

- The length of the interval from $L(s)$ to $U(s)$ is the height of $\mathcal{F}^s$ which is at most $\Delta_{clf}(d-1, n-1)$.
- Before $L(s)$ and after $U(s)$ there are at most $\lfloor n/2 \rfloor$ active symbols.
- The respective intervals may be viewed as CLFs with at most $\lfloor n/2 \rfloor$ symbols, which have heights at most $\Delta_{clf}(d, \lfloor n/2 \rfloor)$.

# Quasi-polynomial upper bound




- We get: $\Delta_{clf}(d, n) \;\leq\; \Delta_{clf}(d-1, n-1) + 2\Delta_{clf}(d, \lfloor n/2 \rfloor)$

# Quasi-polynomial upper bound

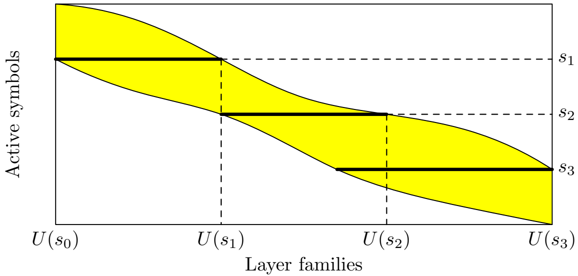



- We get: $\Delta_{clf}(d,n) \leq \Delta_{clf}(d-1,n-1) + 2\Delta_{clf}(d, \lfloor n/2 \rfloor)$
- Using $\Delta_{clf}(1,n) = n$ and $\Delta_{clf}(d,n) = 0$ for $d > n$, the following theorem is proved by induction:

**Theorem (Kalai and Kleitman (1992))**

$\Delta_{clf}(d,n) \leq n^{\log d + 1}$.

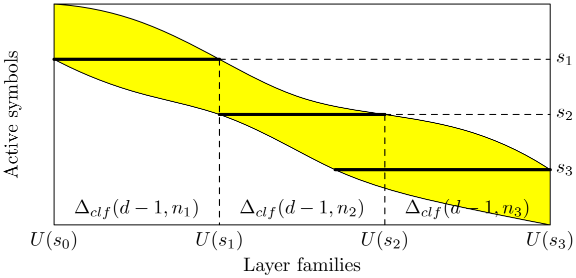
# Better bound for small $d$




- Define $U(s_0) := 0$, and pick a maximal sequence of symbols $s_1, s_2, \ldots, s_k$ such that:

$$s_{i+1} = \operatorname*{argmax}_s \{U(s) \mid L(s) \leq U(s_i) + 1\}$$

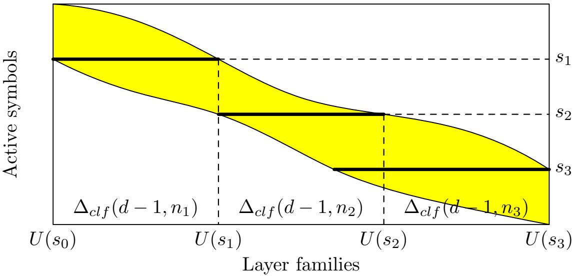
# Better bound for small $d$




- Let $n_i$ be the number of active symbols in the interval $[U(s_{i-1})+1, U(s_i)]$, then:

$$\Delta_{clf}(d, n) \leq \sum_{i=1}^{k} \Delta_{clf}(d-1, n_i)$$

# Better bound for small $d$




- Let $n_i$ be the number of active symbols in the interval $[U(s_{i-1})+1, U(s_i)]$, then:

$$\Delta_{clf}(d,n) \leq \sum_{i=1}^{k} \Delta_{clf}(d-1,n_i)$$

- Each symbol appears in at most 2 intervals: $\sum_{i=1}^{k} n_i \leq 2n$.

# Better bound for small $d$

**Theorem (Larman (1970))**

$\Delta_{clf}(d,n) \le 2^{d-1} n.$

**Proof:**

- By induction:

$$\Delta_{clf}(d,n) \le \sum_{i=1}^{k} \Delta_{clf}(d-1,n_i) \le \sum_{i=1}^{k} 2^{d-2} n_i$$

$$= 2^{d-2} \sum_{i=1}^{k} n_i \le 2^{d-2} 2n = 2^{d-1} n$$

- The presented upper bounds hold even when the layer families contain multisets. I.e., $\{1, 1, 2\} \cap \{1, 2, 3\} = \{1, 2\}$.

# Polymath3: A good place to start

- The presented upper bounds hold even when the layer families contain multisets. I.e., $\{1,1,2\} \cap \{1,2,3\} = \{1,2\}$.
- Let $\Delta^m_{clf}(d,n)$ be the maximum height of any $d$-dimensional connected layer family with multisets using $n$ symbols.

# Polymath3: A good place to start

- The presented upper bounds hold even when the layer families contain multisets. I.e., $\{1,1,2\} \cap \{1,2,3\} = \{1,2\}$.
- Let $\Delta^m_{clf}(d,n)$ be the maximum height of any $d$-dimensional connected layer family with multisets using $n$ symbols.
- Note that $\Delta(d,n) \leq \Delta_{clf}(d,n) + 1 \leq \Delta^m_{clf}(d,n) + 1$.

# Polymath3: A good place to start

- The presented upper bounds hold even when the layer families contain multisets. I.e., $\{1,1,2\} \cap \{1,2,3\} = \{1,2\}$.
- Let $\Delta^m_{clf}(d,n)$ be the maximum height of any $d$-dimensional connected layer family with multisets using $n$ symbols.
- Note that $\Delta(d,n) \leq \Delta_{clf}(d,n) + 1 \leq \Delta^m_{clf}(d,n) + 1$.
- It is not difficult to show that $\Delta^m_{clf}(d,n) \geq d(n-1)+1$:

$$\{1,1,1\},\{1,1,2\},\{1,2,2\},\{2,2,2\},\{2,2,3\},\{2,3,3\},\dots$$

# Polymath3: A good place to start

- The presented upper bounds hold even when the layer families contain multisets. I.e., $\{1,1,2\} \cap \{1,2,3\} = \{1,2\}$.
- Let $\Delta^m_{clf}(d,n)$ be the maximum height of any $d$-dimensional connected layer family with multisets using $n$ symbols.
- Note that $\Delta(d,n) \leq \Delta_{clf}(d,n) + 1 \leq \Delta^m_{clf}(d,n) + 1$.
- It is not difficult to show that $\Delta^m_{clf}(d,n) \geq d(n-1)+1$:

$$\{1,1,1\},\{1,1,2\},\{1,2,2\},\{2,2,2\},\{2,2,3\},\{2,3,3\},\ldots$$

**Conjecture (Hähnle (polymath3))**

$\Delta^m_{clf}(d,n) = d(n-1)+1.$

- Justification: `http://tinyurl.com/3qf556p`

# Polymath3: A good place to start

- The presented upper bounds hold even when the layer families contain multisets. I.e., $\{1,1,2\} \cap \{1,2,3\} = \{1,2\}$.
- Let $\Delta^m_{clf}(d,n)$ be the maximum height of any $d$-dimensional connected layer family with multisets using $n$ symbols.
- Note that $\Delta(d,n) \leq \Delta_{clf}(d,n) + 1 \leq \Delta^m_{clf}(d,n) + 1$.
- It is not difficult to show that $\Delta^m_{clf}(d,n) \geq d(n-1)+1$:

$$\{1,1,1\},\{1,1,2\},\{1,2,2\},\{2,2,2\},\{2,2,3\},\{2,3,3\},\ldots$$

**Conjecture (Hähnle (polymath3))**

$\Delta^m_{clf}(d,n) = d(n-1)+1$.

- Justification: `http://tinyurl.com/3qf556p`
- **Open problem:** Close the gap

$$3(n-1)+1 \ \leq\ \Delta^m_{clf}(3,n) \ \leq\ 4n.$$

# Overview

- **Lecture 1:**
  - Introduction to linear programming and the simplex algorithm.
  - Pivoting rules.
  - The RandomFacet pivoting rule.
- **Lecture 2:**
  - The Hirsch conjecture.
  - Introduction to Markov decision processes (MDPs).
  - Upper bound for the LargestCoefficient pivoting rule for MDPs.
- **Lecture 3:**
  - Lower bounds for pivoting rules utilizing MDPs. Example: Bland's rule.
  - Lower bound for the RandomEdge pivoting rule.
  - Abstractions and related problems.

- Solving **Markov decision processes** (MDPs) is an important problem in *operations research* and *machine learning*; it is, for instance, used to solve the *dairy cow replacement problem*.

# Markov decision processes

- Markov decision processes (MDPs) is a special case of Shapley's stochastic games (1953). They were introduced by Bellman (1957).

# Markov decision processes

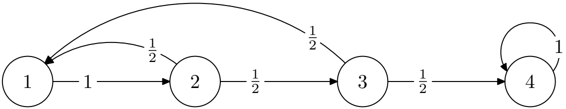
- Markov decision processes (MDPs) is a special case of Shapley's stochastic games (1953). They were introduced by Bellman (1957).
- MDPs can be solved by linear programming, and also **solving MDPs in strongly polynomial time remains open**.

# Markov decision processes

- Markov decision processes (MDPs) is a special case of Shapley's stochastic games (1953). They were introduced by Bellman (1957).
- MDPs can be solved by linear programming, and also **solving MDPs in strongly polynomial time remains open**.
- Ye (2010) showed that the simplex algorithm with the LargestCoefficient pivoting rule solves **discounted** MDPs with a *fixed* discount factor in strongly polynomial time.

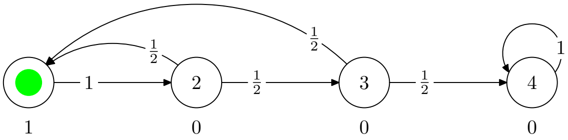
- Markov decision processes (MDPs) is a special case of Shapley's stochastic games (1953). They were introduced by Bellman (1957).
- MDPs can be solved by linear programming, and also **solving MDPs in strongly polynomial time remains open**.
- Ye (2010) showed that the simplex algorithm with the LargestCoefficient pivoting rule solves **discounted** MDPs with a *fixed* discount factor in strongly polynomial time.
- Friedmann, Hansen and Zwick (2011) used MDPs to get lower bounds of subexponential form for the RandomEdge and RandomFacet pivoting rules and the Randomized Bland's rule, and Friedmann (2011) for the LeastEntered pivoting rule.

# Markov chains




$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- An $n$-state **Markov chain** is defined by an $n \times n$ stochastic matrix $P$, with $P_{i,j}$ being the probability of making a transition from state $i$ to state $j$. I.e., $\sum_j P_{i,j} = 1$.

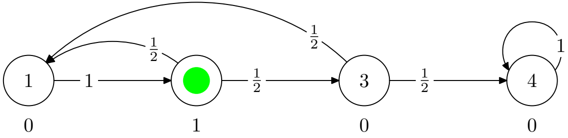



$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- An $n$-state **Markov chain** is defined by an $n \times n$ stochastic matrix $P$, with $P_{i,j}$ being the probability of making a transition from state $i$ to state $j$. I.e., $\sum_j P_{i,j} = 1$.
- Let $b \in \mathbb{R}^n$ define a probability distribution for picking an initial state.
- $b^T P^k$ is a vector defining the probabilities of being in each state after $k$ transitions.

| $k$ | $b^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |

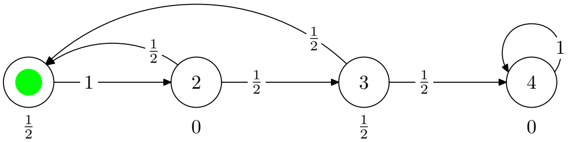
# Markov chains




$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

| $k$ | $b^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |

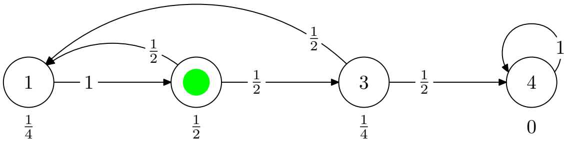
- An $n$-state **Markov chain** is defined by an $n \times n$ stochastic matrix $P$, with $P_{i,j}$ being the probability of making a transition from state $i$ to state $j$. I.e., $\sum_j P_{i,j} = 1$.
- Let $b \in \mathbb{R}^n$ define a probability distribution for picking an initial state.
- $b^T P^k$ is a vector defining the probabilities of being in each state after $k$ transitions.




$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- An $n$-state **Markov chain** is defined by an $n \times n$ stochastic matrix $P$, with $P_{i,j}$ being the probability of making a transition from state $i$ to state $j$. I.e., $\sum_j P_{i,j} = 1$.
- Let $b \in \mathbb{R}^n$ define a probability distribution for picking an initial state.
- $b^T P^k$ is a vector defining the probabilities of being in each state after $k$ transitions.

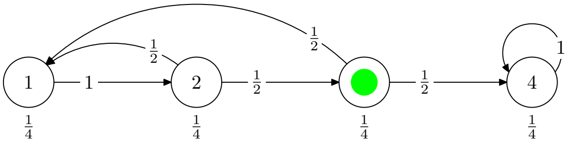
| $k$ | $b^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 2 | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | 0 |




$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- An $n$-state **Markov chain** is defined by an $n \times n$ stochastic matrix $P$, with $P_{i,j}$ being the probability of making a transition from state $i$ to state $j$. I.e., $\sum_j P_{i,j} = 1$.
- Let $b \in \mathbb{R}^n$ define a probability distribution for picking an initial state.
- $b^T P^k$ is a vector defining the probabilities of being in each state after $k$ transitions.

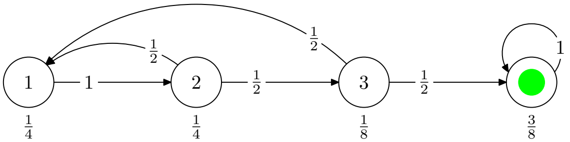
| $k$ | $b^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 2 | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | 0 |
| 3 | $\frac{1}{4}$ | $\frac{1}{2}$ | 0 | $\frac{1}{4}$ |




$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- An $n$-state **Markov chain** is defined by an $n \times n$ stochastic matrix $P$, with $P_{i,j}$ being the probability of making a transition from state $i$ to state $j$. I.e., $\sum_j P_{i,j} = 1$.
- Let $b \in \mathbb{R}^n$ define a probability distribution for picking an initial state.
- $b^T P^k$ is a vector defining the probabilities of being in each state after $k$ transitions.

| $k$ | $b^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 2 | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | 0 |
| 3 | $\frac{1}{4}$ | $\frac{1}{2}$ | 0 | $\frac{1}{4}$ |
| 4 | $\frac{1}{4}$ | $\frac{1}{4}$ | $\frac{1}{4}$ | $\frac{1}{4}$ |




$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

| $k$ | $b^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 2 | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | 0 |
| 3 | $\frac{1}{4}$ | $\frac{1}{2}$ | 0 | $\frac{1}{4}$ |
| 4 | $\frac{1}{4}$ | $\frac{1}{4}$ | $\frac{1}{4}$ | $\frac{1}{4}$ |
| 5 | $\frac{1}{4}$ | $\frac{1}{4}$ | $\frac{1}{8}$ | $\frac{3}{8}$ |
| $\vdots$ | $\vdots$ | | | |

- An $n$-state **Markov chain** is defined by an $n \times n$ stochastic matrix $P$, with $P_{i,j}$ being the probability of making a transition from state $i$ to state $j$. I.e., $\sum_j P_{i,j} = 1$.
- Let $b \in \mathbb{R}^n$ define a probability distribution for picking an initial state.
- $b^T P^k$ is a vector defining the probabilities of being in each state after $k$ transitions.

# Markov chains with rewards

- We refer to the act of leaving a state as an **action**.
- A Markov chain with rewards is a Markov chain $P \in \mathbb{R}^{n \times n}$ where a vector $c \in \mathbb{R}^n$ associates actions with **rewards** (or **costs**). I.e., $c_i$ is the reward for leaving state $i$.
- We are interested in the expected **total reward**, $\sum_{k=0}^{\infty} b^T P^k c$, accumulated for some initial vector $b$. Note that this series generally does not converge.

- We refer to the act of leaving a state as an **action**.
- A Markov chain with rewards is a Markov chain $P \in \mathbb{R}^{n \times n}$ where a vector $c \in \mathbb{R}^n$ associates actions with **rewards** (or **costs**). I.e., $c_i$ is the reward for leaving state $i$.
- We are interested in the expected **total reward**, $\sum_{k=0}^{\infty} b^T P^k c$, accumulated for some initial vector $b$. Note that this series generally does not converge.
- To ensure convergence we introduce a **discount factor** $\gamma < 1$, such that after each transition the Markov chain is stopped with probability $1 - \gamma$. I.e., $(\gamma P)^k \to 0$ for $k \to \infty$.

# Markov chains with rewards

- We refer to the act of leaving a state as an **action**.
- A Markov chain with rewards is a Markov chain $P \in \mathbb{R}^{n \times n}$ where a vector $c \in \mathbb{R}^n$ associates actions with **rewards** (or **costs**). I.e., $c_i$ is the reward for leaving state $i$.
- We are interested in the expected **total reward**, $\sum_{k=0}^{\infty} b^T P^k c$, accumulated for some initial vector $b$. Note that this series generally does not converge.
- To ensure convergence we introduce a **discount factor** $\gamma < 1$, such that after each transition the Markov chain is stopped with probability $1 - \gamma$. I.e., $(\gamma P)^k \to 0$ for $k \to \infty$.
- The expected **total discounted reward** for some $b \in \mathbb{R}^n$ is then $\sum_{k=0}^{\infty} b^T (\gamma P)^k c$.

- Observe that:

$$I = \lim_{\ell \to \infty} I - (\gamma P)^{\ell} = \lim_{\ell \to \infty} (I - \gamma P) \sum_{k=0}^{\ell-1} (\gamma P)^k = (I - \gamma P) \sum_{k=0}^{\infty} (\gamma P)^k$$

- Observe that:

$$I = \lim_{\ell \to \infty} I - (\gamma P)^{\ell} = \lim_{\ell \to \infty} (I - \gamma P) \sum_{k=0}^{\ell-1} (\gamma P)^k = (I - \gamma P) \sum_{k=0}^{\infty} (\gamma P)^k$$

- I.e., $(I - \gamma P)^{-1} = \sum_{k=0}^{\infty} (\gamma P)^k$.

- Proof that $(I - \gamma P)$ is invertible:
  - Assume there is a non-zero linear combination of the columns that equals the zero vector, and let $i$ be the column with largest weight.
  - The $i$'th row cannot sum to zero since the contribution from the diagonal element is numerically larger than the sum of the remaining elements: A contradiction.

# The value vector

- The expected **total discounted reward** for some $b \in \mathbb{R}^n$ is

$$\sum_{k=0}^{\infty} b^T (\gamma P)^k c = b^T (I - \gamma P)^{-1} c.$$

# The value vector

- The expected **total discounted reward** for some $b \in \mathbb{R}^n$ is

$$\sum_{k=0}^{\infty} b^T (\gamma P)^k c = b^T (I - \gamma P)^{-1} c.$$

- We define the **value** of state $i$ as the expected total discounted reward when starting in state $i$ with probability 1:

$$v_i = e_i^T (I - \gamma P)^{-1} c$$

# The value vector

- The expected **total discounted reward** for some $b \in \mathbb{R}^n$ is

$$\sum_{k=0}^{\infty} b^T (\gamma P)^k c = b^T (I - \gamma P)^{-1} c.$$

- We define the **value** of state $i$ as the expected total discounted reward when starting in state $i$ with probability 1:

$$v_i = e_i^T (I - \gamma P)^{-1} c$$

- In general $e_i^T A$ is just the $i$'th row of $A$, and we can define the vector of values $v \in \mathbb{R}^n$ as:

$$v = (I - \gamma P)^{-1} c$$

# The flux vector

- Let $e$ be a vector of ones. Note that the sum of values $e^T v = e^T (I - \gamma P)^{-1} c$ corresponds to setting $b = e$.

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

| $k$ | $e^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | $\frac{1}{2}$ | $\frac{3}{2}$ |
| 2 | $\frac{3}{4}$ | 1 | $\frac{1}{2}$ | $\frac{7}{4}$ |
| 3 | $\frac{3}{4}$ | $\frac{3}{4}$ | $\frac{1}{2}$ | 2 |
| 4 | $\frac{5}{8}$ | $\frac{3}{4}$ | $\frac{3}{8}$ | $\frac{9}{4}$ |
| 5 | $\frac{9}{16}$ | $\frac{5}{8}$ | $\frac{3}{8}$ | $\frac{39}{16}$ |
| $\vdots$ | | $\vdots$ | | |

# The flux vector

- Let $e$ be a vector of ones. Note that the sum of values $e^T v = e^T(I - \gamma P)^{-1} c$ corresponds to setting $b = e$.
- Let $x_i$ be the expected (discounted) number of times action $i$, leaving state $i$, is used for $b = e$. I.e., $x_i$ is the discounted sum of column $i$ in the table.

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

| $k$ | $e^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | $\frac{1}{2}$ | $\frac{3}{2}$ |
| 2 | $\frac{3}{4}$ | 1 | $\frac{1}{2}$ | $\frac{7}{4}$ |
| 3 | $\frac{3}{4}$ | $\frac{3}{4}$ | $\frac{1}{2}$ | 2 |
| 4 | $\frac{5}{8}$ | $\frac{3}{4}$ | $\frac{3}{8}$ | $\frac{9}{4}$ |
| 5 | $\frac{9}{16}$ | $\frac{5}{8}$ | $\frac{3}{8}$ | $\frac{39}{16}$ |
| $\vdots$ | | $\vdots$ | | |

# The flux vector

- Let $e$ be a vector of ones. Note that the sum of values $e^T v = e^T (I - \gamma P)^{-1} c$ corresponds to setting $b = e$.
- Let $x_i$ be the expected (discounted) number of times action $i$, leaving state $i$, is used for $b = e$. I.e., $x_i$ is the discounted sum of column $i$ in the table.
- Equivalently, $x_i$ is the sum of values when $c = e_i$:

$$x_i = e^T (I - \gamma P)^{-1} e_i$$

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

| $k$ | $e^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | $\frac{1}{2}$ | $\frac{3}{2}$ |
| 2 | $\frac{3}{4}$ | 1 | $\frac{1}{2}$ | $\frac{7}{4}$ |
| 3 | $\frac{3}{4}$ | $\frac{3}{4}$ | $\frac{1}{2}$ | 2 |
| 4 | $\frac{5}{8}$ | $\frac{3}{4}$ | $\frac{3}{8}$ | $\frac{9}{4}$ |
| 5 | $\frac{9}{16}$ | $\frac{5}{8}$ | $\frac{3}{8}$ | $\frac{39}{16}$ |
| ⋮ | ⋮ | | | |

# The flux vector

- Let $e$ be a vector of ones. Note that the sum of values $e^T v = e^T(I - \gamma P)^{-1} c$ corresponds to setting $b = e$.
- Let $x_i$ be the expected (discounted) number of times action $i$, leaving state $i$, is used for $b = e$. I.e., $x_i$ is the discounted sum of column $i$ in the table.
- Equivalently, $x_i$ is the sum of values when $c = e_i$:

$$x_i = e^T(I - \gamma P)^{-1} e_i$$

- Hence, we define the **flux vector** $x \in \mathbb{R}^n$ as:

$$x^T = e^T(I - \gamma P)^{-1}$$

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

| $k$ | $e^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | $\frac{1}{2}$ | $\frac{3}{2}$ |
| 2 | $\frac{3}{4}$ | 1 | $\frac{1}{2}$ | $\frac{7}{4}$ |
| 3 | $\frac{3}{4}$ | $\frac{3}{4}$ | $\frac{1}{2}$ | 2 |
| 4 | $\frac{5}{8}$ | $\frac{3}{4}$ | $\frac{3}{8}$ | $\frac{9}{4}$ |
| 5 | $\frac{9}{16}$ | $\frac{5}{8}$ | $\frac{3}{8}$ | $\frac{39}{16}$ |
| $\vdots$ | $\vdots$ | | | |

# The flux vector

- Note that using the flux vector gives a different way of summing up the values:

$$e^T v = c^T x$$

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

| $k$ | $e^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | $\frac{1}{2}$ | $\frac{3}{2}$ |
| 2 | $\frac{3}{4}$ | 1 | $\frac{1}{2}$ | $\frac{7}{4}$ |
| 3 | $\frac{3}{4}$ | $\frac{3}{4}$ | $\frac{1}{2}$ | 2 |
| 4 | $\frac{5}{8}$ | $\frac{3}{4}$ | $\frac{3}{8}$ | $\frac{9}{4}$ |
| 5 | $\frac{9}{16}$ | $\frac{5}{8}$ | $\frac{3}{8}$ | $\frac{39}{16}$ |
| $\vdots$ | $\vdots$ | | | |

# The flux vector

- Note that using the flux vector gives a different way of summing up the values:

$$e^T v = c^T x$$

- Also, each row in the table sums to $n$, meaning that the discounted sum is:

$$e^T x = \frac{n}{1-\gamma}$$

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

| $k$ | $e^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | $\frac{1}{2}$ | $\frac{3}{2}$ |
| 2 | $\frac{3}{4}$ | 1 | $\frac{1}{2}$ | $\frac{7}{4}$ |
| 3 | $\frac{3}{4}$ | $\frac{3}{4}$ | $\frac{1}{2}$ | 2 |
| 4 | $\frac{5}{8}$ | $\frac{3}{4}$ | $\frac{3}{8}$ | $\frac{9}{4}$ |
| 5 | $\frac{9}{16}$ | $\frac{5}{8}$ | $\frac{3}{8}$ | $\frac{39}{16}$ |
| $\vdots$ | $\vdots$ | | | |

# The flux vector

- Note that using the flux vector gives a different way of summing up the values:

$$e^T v = c^T x$$

- Also, each row in the table sums to $n$, meaning that the discounted sum is:

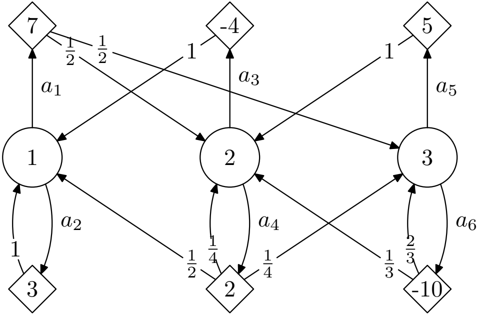
$$e^T x = \frac{n}{1-\gamma}$$

- Finally, $x_i \geq 1$, for all $i$, due to the first row of the table.

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

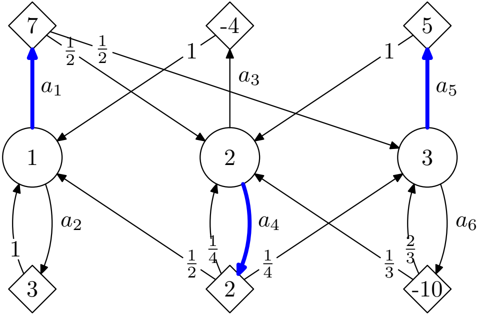
| $k$ | $e^T P^k$ | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | $\frac{1}{2}$ | $\frac{3}{2}$ |
| 2 | $\frac{3}{4}$ | 1 | $\frac{1}{2}$ | $\frac{7}{4}$ |
| 3 | $\frac{3}{4}$ | $\frac{3}{4}$ | $\frac{1}{2}$ | 2 |
| 4 | $\frac{5}{8}$ | $\frac{3}{4}$ | $\frac{3}{8}$ | $\frac{9}{4}$ |
| 5 | $\frac{9}{16}$ | $\frac{5}{8}$ | $\frac{3}{8}$ | $\frac{39}{16}$ |
| $\vdots$ | $\vdots$ | | | |

# Markov decision processes



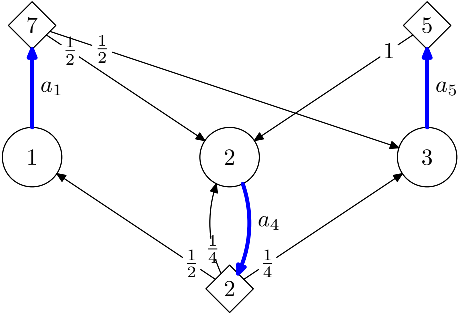

- A Markov decision process consists of a set of $n$ states $S$, each state $i \in S$ being associated with a non-empty set of actions $A_i$.
- Each action $a$ is associated with a reward $c_a$ and a probability distribution $P_a \in \mathbb{R}^{1 \times n}$ such that $P_{a,j}$ is the probability of moving to state $j$ when using action $a$.

# Markov decision processes



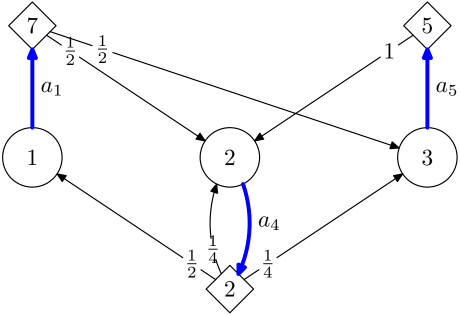

- A **policy** $\pi$ is a choice of an action from each state.

# Markov decision processes



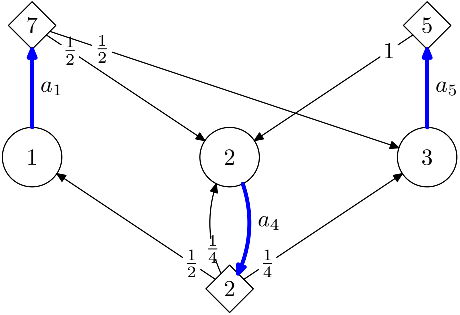

- A **policy** $\pi$ is a choice of an action from each state.
- A policy $\pi$ is a Markov chain with rewards.

# Markov decision processes

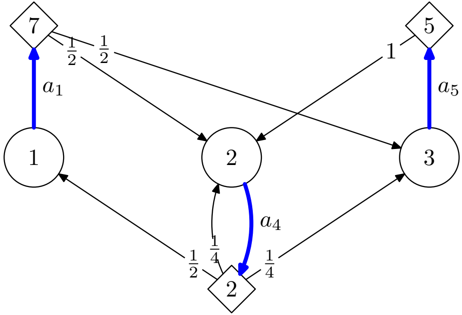



- A **policy** $\pi$ is a choice of an action from each state.
- A policy $\pi$ is a Markov chain with rewards.
- Let $v_\pi$ be the value vector for $\pi$.

# Markov decision processes

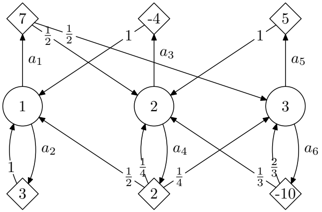



- A **policy** $\pi$ is a choice of an action from each state.
- A policy $\pi$ is a Markov chain with rewards.
- Let $v_\pi$ be the value vector for $\pi$.
- A policy $\pi^*$ is **optimal** if it maximizes the values of all states. I.e., $v_{\pi^*} \geq v_\pi$ for all $\pi$.




- Shapley (1953), Bellman (1957): There always exists an optimal policy.
- Solving an MDP means finding an optimal policy.

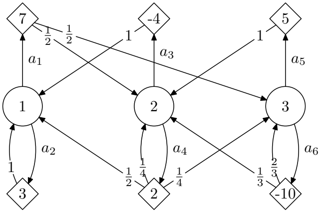
# Markov decision processes




$$J = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad P = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & 0 \\ 0 & \frac{1}{3} & \frac{2}{3} \end{bmatrix} \qquad c = \begin{bmatrix} 7 \\ 3 \\ -4 \\ 2 \\ 5 \\ -10 \end{bmatrix}$$

- A discounted MDP with $n$ states and a total of $m$ actions can be represented by:
  - A discount factor $\gamma < 1$.
  - A zero-one matrix $J \in \{0,1\}^{m \times n}$, with $J_{a,i} = 1$ iff $a \in A_i$.
  - A stochastic matrix $P \in \mathbb{R}^{m \times n}$.
  - A reward vector $c \in \mathbb{R}^m$.

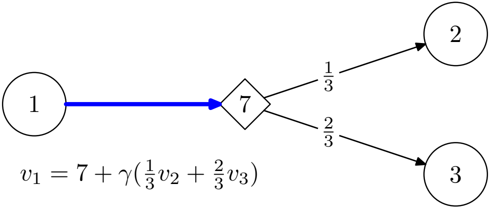
# Markov decision processes




$$J = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad P = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & 0 \\ 0 & \frac{1}{3} & \frac{2}{3} \end{bmatrix} \qquad c = \begin{bmatrix} 7 \\ 3 \\ -4 \\ 2 \\ 5 \\ -10 \end{bmatrix}$$

- A discounted MDP with $n$ states and a total of $m$ actions can be represented by:
  - A discount factor $\gamma < 1$.
  - A zero-one matrix $J \in \{0,1\}^{m \times n}$, with $J_{a,i} = 1$ iff $a \in A_i$.
  - A stochastic matrix $P \in \mathbb{R}^{m \times n}$.
  - A reward vector $c \in \mathbb{R}^m$.
- For some policy $\pi$, $P_\pi$ and $c_\pi$ are obtained by combining the corresponding $n$ rows of $P$ and $c$. Note that $J_\pi = I$.

# The value defining equations




$$v_1 = 7 + \gamma(\tfrac{1}{3}v_2 + \tfrac{2}{3}v_3)$$

- Take a look at the equations defining the value vector $v_\pi$ for some policy $\pi$:

$$v_\pi = (I - \gamma P_\pi)^{-1} c_\pi \quad \iff \quad v_\pi = c_\pi + \gamma P_\pi v_\pi$$

- I.e., the values should be consistent when taking one step.

# Optimal values

- Intuitively, an optimal policy $\pi^*$ must maximize the values locally by using the best actions given the value vector $v_{\pi^*}$.

# Optimal values

- Intuitively, an optimal policy $\pi^*$ must maximize the values locally by using the best actions given the value vector $v_{\pi^*}$.
- Indeed, there exists a unique optimal value vector $v^* \in \mathbb{R}^n$ satisfying:

$$\forall i \in S: \quad v_i^* = \max_{a \in A_i} \; c_a + \gamma P_a v^*$$

- A policy $\pi^*$ is optimal if and only if $v_{\pi^*} = v^*$.

# Optimal values

- Intuitively, an optimal policy $\pi^*$ must maximize the values locally by using the best actions given the value vector $v_{\pi^*}$.
- Indeed, there exists a unique optimal value vector $v^* \in \mathbb{R}^n$ satisfying:

$$\forall i \in S : \quad v_i^* = \max_{a \in A_i} \; c_a + \gamma P_a v^*$$

- A policy $\pi^*$ is optimal if and only if $v_{\pi^*} = v^*$.
- Knowing $v^*$ we can easily construct an optimal policy $\pi^*$ by picking locally optimal actions:

$$\forall i \in S : \quad \pi^*(i) \in \operatorname*{argmax}_{a \in A_i} \; c_a + \gamma P_a v^*$$

# Linear program for solving MDPs

- Standard trick:

$$\max\{a, b\} \quad = \quad \min \; c \;\; \text{s.t.} \;\; c \geq a \;\; \text{and} \;\; c \geq b$$

# Linear program for solving MDPs

- Standard trick:

$$\max\{a, b\} \quad = \quad \min \; c \;\; \text{s.t.} \;\; c \geq a \;\; \text{and} \;\; c \geq b$$

- The requirement:

$$\forall i \in S : \quad v_i^* = \max_{a \in A_i} \; c_a + \gamma P_a v^*$$

can be equivalently stated as $v^*$ being the optimal solution to the linear program:

$$\begin{aligned} &\min_{y \in \mathbb{R}^n} \quad e^T y \\ &s.t. \;\; \forall i \in S, \forall a \in A_i : \quad y_i \geq c_a + \gamma P_a y \end{aligned}$$

# Linear program for solving MDPs

- Standard trick:

$$\max\{a, b\} \quad = \quad \min \ c \ \text{ s.t. } \ c \geq a \ \text{ and } \ c \geq b$$

- The requirement:

$$\forall i \in S: \quad v_i^* = \max_{a \in A_i} \ c_a + \gamma P_a v^*$$

can be equivalently stated as $v^*$ being the optimal solution to the linear program:

$$\begin{aligned} \min_{y \in \mathbb{R}^n} \quad & e^T y \\ s.t. \quad Jy \ & \geq \ c + \gamma P y \end{aligned}$$

# Linear program for solving MDPs

- Standard trick:

$$\max\{a, b\} \quad = \quad \min \; c \;\; \text{s.t.} \;\; c \geq a \;\; \text{and} \;\; c \geq b$$

- The requirement:

$$\forall i \in S : \quad v_i^* = \max_{a \in A_i} \; c_a + \gamma P_a v^*$$

can be equivalently stated as $v^*$ being the optimal solution to the linear program:

$$\begin{aligned} \min_{y \in \mathbb{R}^n} \quad & e^T y \\ s.t. \;\; (J - \gamma P)y \;\; & \geq \;\; c \end{aligned}$$

# Primal and dual LPs for MDPs

$$(P) \quad \begin{array}{lrcl} \max & c^T x & & \\ s.t. & (J - \gamma P)^T x & = & e \\ & x & \geq & 0 \end{array} \qquad (D) \quad \begin{array}{lrcl} \min & e^T y & & \\ s.t. & (J - \gamma P) y & \geq & c \end{array}$$

# Primal and dual LPs for MDPs

$$(P) \quad \begin{array}{lrcl} \max & c^T x & & \\ s.t. & (J-\gamma P)^T x & = & e \\ & x & \geq & 0 \end{array} \qquad (D) \quad \begin{array}{lrcl} \min & e^T y & & \\ s.t. & (J-\gamma P) y & \geq & c \end{array}$$

- Let's take a closer look at the constraints of $(P)$:

$$\forall i \in S: \quad \sum_{a \in A_i} x_a = 1 + \gamma \sum_{j \in S} \sum_{b \in A_j} P_{b,i} x_b$$

# Primal and dual LPs for MDPs

$$(P)\quad \begin{array}{lrcl} \max & c^T x & & \\ s.t. & (J-\gamma P)^T x & = & e \\ & x & \geq & 0 \end{array} \qquad (D)\quad \begin{array}{lrcl} \min & e^T y & & \\ s.t. & (J-\gamma P) y & \geq & c \end{array}$$

- Let's take a closer look at the constraints of $(P)$:

$$\forall i \in S: \quad \sum_{a \in A_i} x_a = 1 + \gamma \sum_{j \in S} \sum_{b \in A_j} P_{b,i} x_b$$

- Since all variables are non-negative the right-hand-side is positive, and at least one variable $x_a$ for $a \in A_i$ is positive for every $i \in S$.

# Primal and dual LPs for MDPs

$$(P)\quad \begin{array}{lrcl} \max & c^T x & & \\ s.t. & (J-\gamma P)^T x & = & e \\ & x & \geq & 0 \end{array} \qquad (D)\quad \begin{array}{lrcl} \min & e^T y & & \\ s.t. & (J-\gamma P) y & \geq & c \end{array}$$

- Let's take a closer look at the constraints of $(P)$:

$$\forall i \in S: \quad \sum_{a \in A_i} x_a = 1 + \gamma \sum_{j \in S} \sum_{b \in A_j} P_{b,i} x_b$$

- Since all variables are non-negative the right-hand-side is positive, and at least one variable $x_a$ for $a \in A_i$ is positive for every $i \in S$.
- In any basic feasible solution $x_B$ with basis $B$ at most $n = |S|$ variables are non-zero.

# Primal and dual LPs for MDPs

$$(P)\quad \begin{array}{lrcl} \max & c^T x & & \\ s.t. & (J-\gamma P)^T x & = & e \\ & x & \geq & 0 \end{array} \qquad (D)\quad \begin{array}{lrcl} \min & e^T y & & \\ s.t. & (J-\gamma P) y & \geq & c \end{array}$$

- Let's take a closer look at the constraints of $(P)$:

$$\forall i \in S : \quad \sum_{a \in A_i} x_a = 1 + \gamma \sum_{j \in S} \sum_{b \in A_j} P_{b,i} x_b$$

- Since all variables are non-negative the right-hand-side is positive, and at least one variable $x_a$ for $a \in A_i$ is positive for every $i \in S$.
- In any basic feasible solution $x_B$ with basis $B$ at most $n = |S|$ variables are non-zero.
- There must be exactly one positive variable for each state, and $B$ can be interpreted as a policy $\pi$.

# Policies and basic feasible solutions

$$
(P) \quad \begin{array}{lrcl} \max & c^T x & & \\ s.t. & (J - \gamma P)^T x & = & e \\ & x & \geq & 0 \end{array}
$$

- Recall that $(I - \gamma P_{\pi})$ is invertible for every policy $\pi$, such that $\pi$ forms a basis for $(P)$.

# Policies and basic feasible solutions

$$(P) \quad \begin{array}{lrcl} \max & c^T x & & \\ s.t. & (J-\gamma P)^T x & = & e \\ & x & \geq & 0 \end{array}$$

- Recall that $(I-\gamma P_{\pi})$ is invertible for every policy $\pi$, such that $\pi$ forms a basis for $(P)$.
- Let $x_{\pi} \in \mathbb{R}^n$ and $x_{\bar{\pi}} \in \mathbb{R}^{m-n}$ be vectors of basic and non-basic variables for $\pi$.
- The basic variables must satisfy $(I-\gamma P_{\pi})^T x_{\pi} = e$, which is exactly the definition of the flux vector for $\pi$.

# Policies and basic feasible solutions

$$(P) \quad \begin{array}{lrcl} \max & c^T x & & \\ s.t. & (J-\gamma P)^T x & = & e \\ & x & \geq & 0 \end{array}$$

- Recall that $(I-\gamma P_\pi)$ is invertible for every policy $\pi$, such that $\pi$ forms a basis for $(P)$.
- Let $x_\pi \in \mathbb{R}^n$ and $x_{\bar{\pi}} \in \mathbb{R}^{m-n}$ be vectors of basic and non-basic variables for $\pi$.
- The basic variables must satisfy $(I-\gamma P_\pi)^T x_\pi = e$, which is exactly the definition of the flux vector for $\pi$.
- Since all variables of a flux vector are greater than 1, $(x_\pi, x_{\bar{\pi}})$ is a basic feasible solution for $(P)$.

# Policies and basic feasible solutions

$$
(P) \quad \begin{array}{lrcl} \max & c^T x & & \\ s.t. & (J - \gamma P)^T x & = & e \\ & x & \geq & 0 \end{array}
$$

- Recall that $(I - \gamma P_{\pi})$ is invertible for every policy $\pi$, such that $\pi$ forms a basis for $(P)$.
- Let $x_{\pi} \in \mathbb{R}^n$ and $x_{\bar{\pi}} \in \mathbb{R}^{m-n}$ be vectors of basic and non-basic variables for $\pi$.
- The basic variables must satisfy $(I - \gamma P_{\pi})^T x_{\pi} = e$, which is exactly the definition of the flux vector for $\pi$.
- Since all variables of a flux vector are greater than 1, $(x_{\pi}, x_{\bar{\pi}})$ is a basic feasible solution for $(P)$.
- Hence, there is a one-to-one correspondence between policies and basic feasible solutions of the primal LP $(P)$.

# The reduced costs

- Let $\pi$ be a basis. The reduced cost vector $\bar{c}^{\pi} \in \mathbb{R}^m$, i.e. the coefficients of the corresponding tableau, is defined as:

$$\bar{c}^{\pi} = c - (J - \gamma P)(I - \gamma P)^{-1} c_{\pi} = c - (J - \gamma P) v_{\pi}$$

- Equivalently, for all $i \in S$ and $a \in A_i$:

$$\bar{c}^{\pi}_a = (c_a + \gamma P_a v_{\pi}) - (v_{\pi})_i$$

- Hence, $\bar{c}^{\pi}_a$ is the improvement over the current value by using $a$ for one step w.r.t. $v_{\pi}$.
- If $\bar{c}^{\pi}_a > 0$ we say that $a$ is an **improving switch**.

# Improving switches and multiple joint pivots

**Lemma (Howard (1960))**

*Let $\pi'$ be obtained from $\pi$ by jointly performing any non-empty set of improving switches. Then $v_{\pi'} \geq v_\pi$ and $v_{\pi'} \neq v_\pi$.*

**Lemma (Howard (1960))**

*A policy $\pi$ is optimal iff there are no improving switches.*

# Policy iteration

**Function** POLICYITERATION($\pi$)

**while** $\exists$ *improving switch w.r.t.* $\pi$ **do**
    Update $\pi$ by performing improving switches

**return** $\pi$

**Function** POLICYITERATION($\pi$)

**while** $\exists$ *improving switch w.r.t.* $\pi$ **do**

  Update $\pi$ by performing improving switches

**return** $\pi$

- Howard's algorithm: Perform as many improving switches as possible. More precisely,

$$\forall i \in S : \quad \pi(i) \leftarrow \operatorname*{argmax}_{a \in A_i} \; \bar{c}_a^{\pi}$$

# The LargestCoefficient pivoting rule for MDPs

**Theorem (Ye (2010))**

*The simplex algorithm with the* LargestCoefficient *pivoting rule solves the primal LP of an n-state MDP with m actions and discount factor* $\gamma < 1$ *in at most* $O(\frac{mn}{1-\gamma} \log \frac{n}{1-\gamma})$ *steps. The same is true for Howard's algorithm.*

# The LargestCoefficient pivoting rule for MDPs

**Theorem (Ye (2010))**

*The simplex algorithm with the* LargestCoefficient *pivoting rule solves the primal LP of an n-state MDP with m actions and discount factor* $\gamma < 1$ *in at most* $O(\frac{mn}{1-\gamma} \log \frac{n}{1-\gamma})$ *steps. The same is true for Howard's algorithm.*

- When $\gamma$ is some fixed constant this gives a strongly polynomial bound. I.e., a polynomial bound only depending on $n$ and $m$.

# The LargestCoefficient pivoting rule for MDPs

**Theorem (Ye (2010))**

*The simplex algorithm with the* LargestCoefficient *pivoting rule solves the primal LP of an n-state MDP with m actions and discount factor $\gamma < 1$ in at most $O(\frac{mn}{1-\gamma} \log \frac{n}{1-\gamma})$ steps. The same is true for Howard's algorithm.*

- When $\gamma$ is some fixed constant this gives a strongly polynomial bound. I.e., a polynomial bound only depending on $n$ and $m$.
- The idea of the proof is to show that for every $O(\frac{n}{1-\gamma} \log \frac{n}{1-\gamma})$ pivoting steps a new variable will never enter the basis again.

# The LargestCoefficient pivoting rule for MDPs

- For some policy $\pi$ with basic feasible solution $(x_\pi, x_{\bar{\pi}})$ the tableau method rewrites the objective function as:

$$\max \quad z + (\bar{c}^{\pi})^T x$$

where $z = c_\pi^T x_\pi = e^T (I - \gamma P_\pi)^{-1} c_\pi$ is the current value, and $\bar{c}^{\pi}$ is the reduced cost vector.

# The LargestCoefficient pivoting rule for MDPs

- For some policy $\pi$ with basic feasible solution $(x_\pi, x_{\bar{\pi}})$ the tableau method rewrites the objective function as:

$$\max \quad z + (\bar{c}^\pi)^T x$$

  where $z = c_\pi^T x_\pi = e^T(I - \gamma P_\pi)^{-1} c_\pi$ is the current value, and $\bar{c}^\pi$ is the reduced cost vector.

- Let $\Delta_{\bar{\pi}} = \max_a \; \bar{c}_a^\pi$ be the largest coefficient.

# The LargestCoefficient pivoting rule for MDPs

- For some policy $\pi$ with basic feasible solution $(x_\pi, x_{\bar{\pi}})$ the tableau method rewrites the objective function as:

$$\max \quad z + (\bar{c}^\pi)^T x$$

where $z = c_\pi^T x_\pi = e^T (I - \gamma P_\pi)^{-1} c_\pi$ is the current value, and $\bar{c}^\pi$ is the reduced cost vector.

- Let $\Delta_{\bar{\pi}} = \max_a \; \bar{c}_a^\pi$ be the largest coefficient.
- The new objective function is equivalent to the original objective function, and in particular the optimal value $z^*$ is upper bounded by the largest conceivable increase:

$$z^* \;\leq\; c_\pi^T x_\pi + \frac{n}{1-\gamma} \Delta_{\bar{\pi}}$$

# The LargestCoefficient pivoting rule for MDPs

- Let $x_a$ be the non-basic variable with coefficient $\Delta_{\bar{\pi}}$ for some policy $\pi$.
- The LargestCoefficient pivoting rule constructs the next basis $\pi'$ by increasing $x_a$ until a basic variable becomes zero.

# The LargestCoefficient pivoting rule for MDPs

- Let $x_a$ be the non-basic variable with coefficient $\Delta_{\bar{\pi}}$ for some policy $\pi$.
- The LargestCoefficient pivoting rule constructs the next basis $\pi'$ by increasing $x_a$ until a basic variable becomes zero.
- The improvement in value $c_{\pi'}^T x_{\pi'} - c_{\pi}^T x_{\pi}$ is the increase in $x_a$ multiplied by $\Delta_{\bar{\pi}}$.

# The LargestCoefficient pivoting rule for MDPs

- Let $x_a$ be the non-basic variable with coefficient $\Delta_{\bar{\pi}}$ for some policy $\pi$.
- The LargestCoefficient pivoting rule constructs the next basis $\pi'$ by increasing $x_a$ until a basic variable becomes zero.
- The improvement in value $c_{\pi'}^T x_{\pi'} - c_{\pi}^T x_{\pi}$ is the increase in $x_a$ multiplied by $\Delta_{\bar{\pi}}$.
- Since $x_a$ is part of the flux vector of $\pi'$, the new value of $x_a$ is at least 1, and we get:

$$c_{\pi'}^T x_{\pi'} - c_{\pi}^T x_{\pi} \geq \Delta_{\bar{\pi}}$$

# The LargestCoefficient pivoting rule for MDPs

- Let $x_a$ be the non-basic variable with coefficient $\Delta_{\bar{\pi}}$ for some policy $\pi$.
- The LargestCoefficient pivoting rule constructs the next basis $\pi'$ by increasing $x_a$ until a basic variable becomes zero.
- The improvement in value $c_{\pi'}^T x_{\pi'} - c_\pi^T x_\pi$ is the increase in $x_a$ multiplied by $\Delta_{\bar{\pi}}$.
- Since $x_a$ is part of the flux vector of $\pi'$, the new value of $x_a$ is at least 1, and we get:

$$c_{\pi'}^T x_{\pi'} - c_\pi^T x_\pi \geq \Delta_{\bar{\pi}}$$

- Howard's algorithm also constructs a new policy containing $x_a$, meaning that this increase is again guaranteed.

# The LargestCoefficient pivoting rule for MDPs

- Let $x_a$ be the non-basic variable with coefficient $\Delta_{\bar{\pi}}$ for some policy $\pi$.
- The LargestCoefficient pivoting rule constructs the next basis $\pi'$ by increasing $x_a$ until a basic variable becomes zero.
- The improvement in value $c_{\pi'}^T x_{\pi'} - c_\pi^T x_\pi$ is the increase in $x_a$ multiplied by $\Delta_{\bar{\pi}}$.
- Since $x_a$ is part of the flux vector of $\pi'$, the new value of $x_a$ is at least 1, and we get:

$$c_{\pi'}^T x_{\pi'} - c_\pi^T x_\pi \geq \Delta_{\bar{\pi}}$$

- Howard's algorithm also constructs a new policy containing $x_a$, meaning that this increase is again guaranteed.
- **Note:** This is the only part of the analysis affected by the chosen pivoting rule. I.e., the proof also works for the LargestIncrease pivoting rule.

- Combining

$$z^* \;\le\; c_\pi^T x_\pi + \frac{n}{1-\gamma}\Delta_{\bar{\pi}} \quad \text{and} \quad c_{\pi'}^T x_{\pi'} - c_\pi^T x_\pi \ge \Delta_{\bar{\pi}}$$

gives

$$z^* \;\le\; c_\pi^T x_\pi + \frac{n}{1-\gamma}(c_{\pi'}^T x_{\pi'} - c_\pi^T x_\pi) \qquad \Longleftrightarrow$$

$$z^* - c_{\pi'}^T x_{\pi'} \;\le\; \left(1 - \frac{1-\gamma}{n}\right)(z^* - c_\pi^T x_\pi)$$

# The LargestCoefficient pivoting rule for MDPs

- Combining

$$z^* \;\leq\; c_\pi^T x_\pi + \frac{n}{1-\gamma}\Delta_{\bar{\pi}} \quad \text{and} \quad c_{\pi'}^T x_{\pi'} - c_\pi^T x_\pi \geq \Delta_{\bar{\pi}}$$

gives

$$z^* \;\leq\; c_\pi^T x_\pi + \frac{n}{1-\gamma}(c_{\pi'}^T x_{\pi'} - c_\pi^T x_\pi) \qquad \iff$$

$$z^* - c_{\pi'}^T x_{\pi'} \;\leq\; \left(1 - \frac{1-\gamma}{n}\right)(z^* - c_\pi^T x_\pi)$$

- Hence, each step brings us significantly closer to the optimal value.

# The LargestCoefficient pivoting rule for MDPs

- Let $\pi^t$ be the basic feasible solution obtained after $t$ pivoting steps, starting from $\pi^0$, then:

$$z^* - c_{\pi^t}^T x_{\pi^t} \;\leq\; \left(1 - \frac{1-\gamma}{n}\right)^t (z^* - c_{\pi^0}^T x_{\pi^0})$$

- The bound is then combined with:[2]

**Lemma**

*Let $\pi^*$, $\pi^t$ and $\pi^0$ be three policies with $v_{\pi^*} \geq v_{\pi^t} \geq v_{\pi^0}$. Let $a = \operatorname{argmax}_{a \in \pi^0} \bar{c}_a^{\pi^*}$, and assume $a \in \pi^t$. Then:*

$$e^T v_{\pi^*} - c_{\pi^t}^T x_{\pi^t} \geq \frac{1-\gamma}{n}(e^T v_{\pi^*} - c_{\pi^0}^T x_{\pi^0})$$

[2] This particular formulation of the lemma is from Hansen, Miltersen and Zwick (2011).

- We get:

$$\frac{1-\gamma}{n} \leq \frac{z^* - c_{\pi^t}^T x_{\pi^t}}{z^* - c_{\pi^0}^T x_{\pi^0}} \leq \left(1 - \frac{1-\gamma}{n}\right)^t$$

- Using $\log(1-x) \leq -x$ for $x < 1$ gives:

$$t \ \leq \ \frac{n}{1-\gamma} \log \frac{n}{1-\gamma}$$

- Hence, after more than $\frac{n}{1-\gamma} \log \frac{n}{1-\gamma}$ steps, the action $a$ specified by the lemma can never enter the basis again, which completes the proof.

# Overview

- **Lecture 1:**
  - Introduction to linear programming and the simplex algorithm.
  - Pivoting rules.
  - The RandomFacet pivoting rule.
- **Lecture 2:**
  - The Hirsch conjecture.
  - Introduction to Markov decision processes (MDPs).
  - Upper bound for the LargestCoefficient pivoting rule for MDPs.
- **Lecture 3:**
  - Lower bounds for pivoting rules utilizing MDPs. Example: Bland's rule.
  - Lower bound for the RandomEdge pivoting rule.
  - Abstractions and related problems.