Lemke's Algorithm For Discounted Games

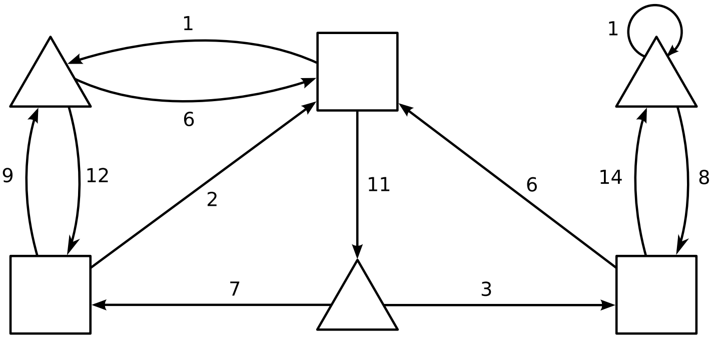
John Fearnley Marcin Jurdziński Rahul Savani

University of Warwick

GASICS Workshop 29th June 2009

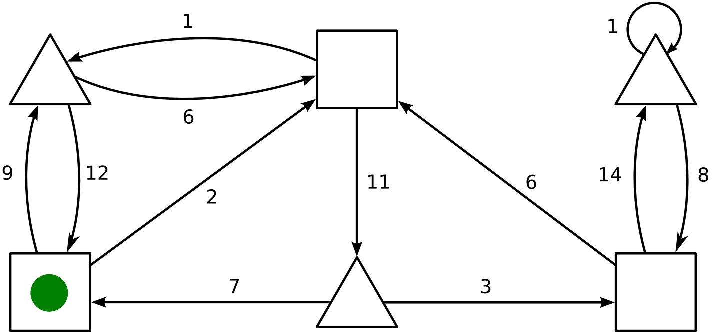
Discounted Games

Discount factor $\beta = 0.5$



Discounted Games

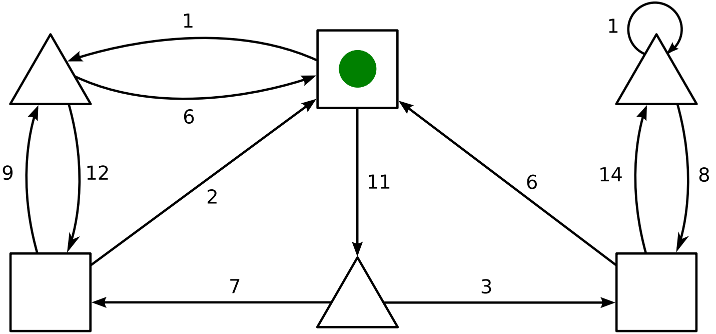
Discount factor $\beta = 0.5$



Payoff $=$

Discounted Games

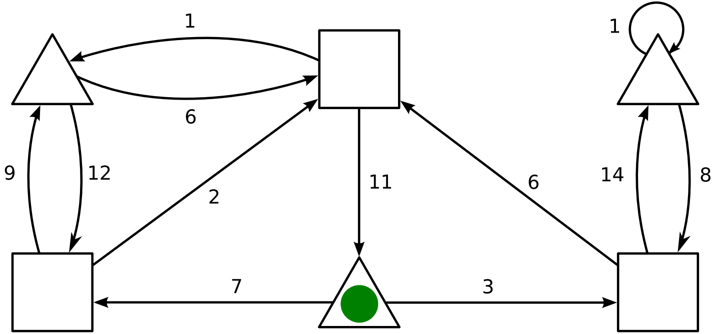
Discount factor $\beta = 0.5$



Payoff $= 2$

Discounted Games

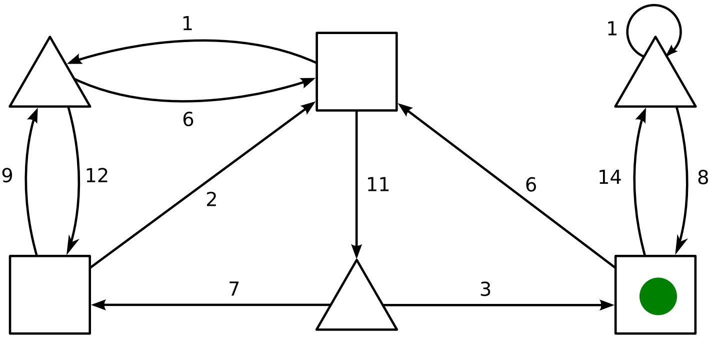
Discount factor $\beta = 0.5$



$$\text{Payoff} = 2 + (\beta \times 11)$$

Discounted Games

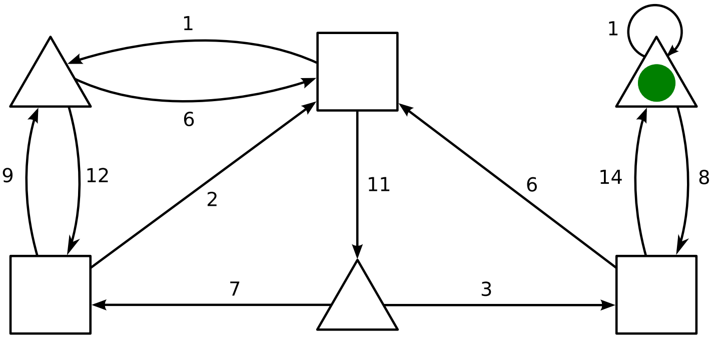
Discount factor $\beta = 0.5$



$$\text{Payoff} = 2 + (\beta \times 11) + (\beta^2 \times 3)$$

Discounted Games

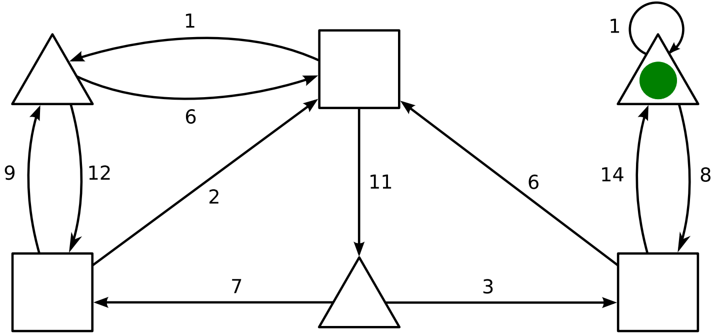
Discount factor $\beta = 0.5$



$$\text{Payoff} = 2 + (\beta \times 11) + (\beta^2 \times 3) + (\beta^3 \times 14)$$

Discounted Games

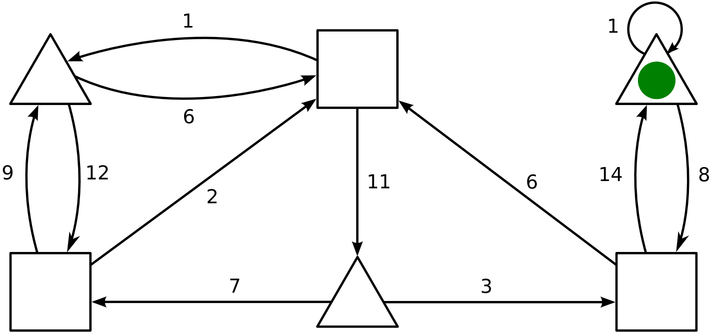
Discount factor $\beta = 0.5$



$$\text{Payoff} = 2 + (\beta \times 11) + (\beta^2 \times 3) + (\beta^3 \times 14) + (\beta^4 \times 1) \ \ldots$$

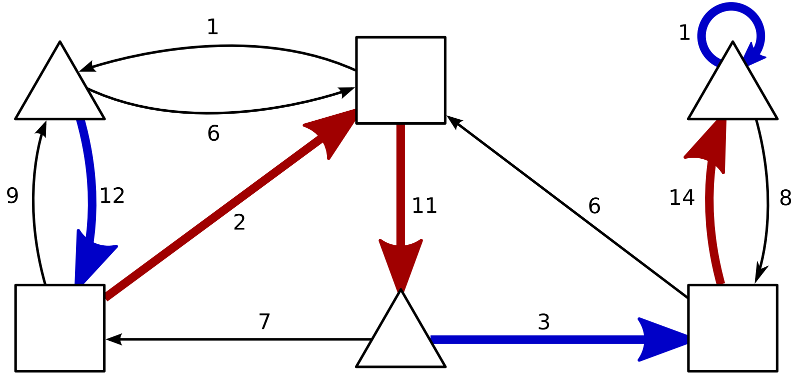
Discounted Games

Discount factor $\beta = 0.5$



$$\text{Payoff} = 2 + (\beta \times 11) + (\beta^2 \times 3) + (\beta^3 \times 14) + (\beta^4 \times 1) \ \ldots$$
$$= \sum_{i=0}^{\infty} \beta^i \cdot \text{reward(i)} = 10.125$$

Positional Strategies



Determinacy

Theorem (Shapley 1953)

For every initial vertex v in a discounted game there is a constant c such that:

- Max has a positional strategy against which the payoff is $\geq c$
- Min has a positional strategy against which the payoff is $\leq c$

Determinacy

Theorem (Shapley 1953)

For every initial vertex v in a discounted game there is a constant c such that:

- Max has a positional strategy against which the payoff is $\geq c$
- Min has a positional strategy against which the payoff is $\leq c$

Computational problem:

- Find c
- Find the two optimal strategies

Motivation

Discounted games are important because:

- Parity games and mean-payoff games can be reduced to them
- Model checking μ-calculus can be reduced to discounted games

Motivation

Discounted games are important because:

- Parity games and mean-payoff games can be reduced to them
- Model checking μ-calculus can be reduced to discounted games
- They are in NP $\cap$ co-NP but no polynomial time algorithm is known

The Linear Complementarity Problem

The linear complementarity problem (LCP) is a fundamental problem in mathematical optimization

Given $M \in \mathbb{R}^{n \times n}$ and $q \in \mathbb{R}^n$, find $w, z \in \mathbb{R}^n$ that satisfy:

$$w = Mz + q$$

The Linear Complementarity Problem

The linear complementarity problem (LCP) is a fundamental problem in mathematical optimization

Given $M \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$, find $w, z \in \mathbb{R}^n$ that satisfy:

$$\begin{aligned} w &= Mz + q \\ w, z &\geq 0 \end{aligned}$$

The Linear Complementarity Problem

The linear complementarity problem (LCP) is a fundamental problem in mathematical optimization

Given $M \in \mathbb{R}^{n \times n}$ and $q \in \mathbb{R}^n$, find $w, z \in \mathbb{R}^n$ that satisfy:

$$\begin{aligned} w &= Mz + q \\ w, z &\geq 0 \\ z_i w_i &= 0 \qquad \forall i \in \{1 \ldots n\} \end{aligned}$$

Lemke's algorithm

Pivoting algorithm given by Lemke in 1965

Lemke's algorithm

Pivoting algorithm given by Lemke in 1965

Modify the LCP with a scalar z_0 and positive covering vector d

$$
\begin{aligned}
w &= Mz + q + dz_0 \\
w, z &\geq 0 \\
z_i w_i &= 0 \qquad\qquad \forall i \in \{1 \ldots n\}
\end{aligned}
$$

Lemke's algorithm

Pivoting algorithm given by Lemke in 1965

Modify the LCP with a scalar z_0 and positive covering vector d

$$
\begin{aligned}
w &= Mz + q + dz_0 \\
w, z &\geq 0 \\
z_i w_i &= 0 \qquad \forall i \in \{1 \ldots n\}
\end{aligned}
$$

When z_0 is large the modified game has a trivial solution

$$
\begin{aligned}
w &= q + dz_0 \\
z &= 0
\end{aligned}
$$

Lemke's algorithm

Pivoting algorithm given by Lemke in 1965

Modify the LCP with a scalar z_0 and positive covering vector d

$$\begin{aligned} w &= Mz + q + dz_0 \\ w, z &\geq 0 \\ z_i w_i &= 0 \qquad \forall i \in \{1 \ldots n\} \end{aligned}$$

When z_0 is large the modified game has a trivial solution

$$\begin{aligned} w &= q + dz_0 \\ z &= 0 \end{aligned}$$

The idea is to drive z_0 down while not violating any constraint

Geometric Interpretation

$$
\begin{aligned}
q &= Iw - Mz \\
w, z &\geq 0 \\
z_i w_i &= 0 \qquad\qquad \forall i \in \{1 \ldots n\}
\end{aligned}
$$

Geometric Interpretation

$$
\begin{aligned}
q &= Iw - Mz \\
w, z &\geq 0 \\
z_i w_i &= 0 \qquad\qquad \forall i \in \{1 \ldots n\}
\end{aligned}
$$

Definition

For every i in $\{1 \ldots \mathrm{n}\}$ we choose the ith column of either $-M$ or I

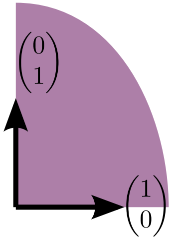
The complementary cone contains all points that are positive combinations of these columns

Geometric Interpretation

$$q = Iw - Mz$$
$$w, z \geq 0$$
$$z_i w_i = 0 \qquad \forall i \in \{1 \dots n\}$$

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} M = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} q = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

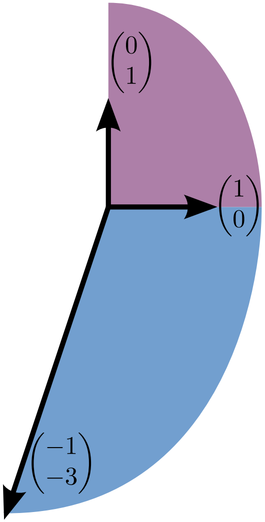
Geometric Interpretation



$$\begin{aligned} q &= Iw - Mz \\ w, z &\geq 0 \\ z_i w_i &= 0 \qquad \forall i \in \{1 \dots n\} \end{aligned}$$

$$I = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} M = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} q = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

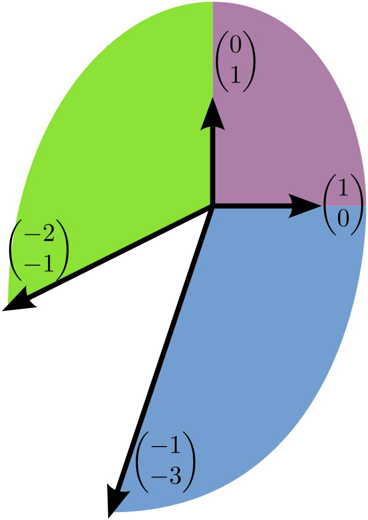
Geometric Interpretation



$$q = Iw - Mz$$
$$w, z \geq 0$$
$$z_i w_i = 0 \qquad \forall i \in \{1 \ldots n\}$$

$$I = \begin{pmatrix} \mathbf{1} & 0 \\ \mathbf{0} & 1 \end{pmatrix} M = \begin{pmatrix} 2 & \mathbf{1} \\ 1 & \mathbf{3} \end{pmatrix} q = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

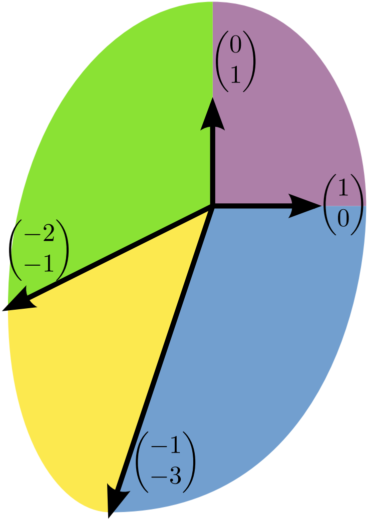
Geometric Interpretation



$$q = Iw - Mz$$
$$w, z \geq 0$$
$$z_i w_i = 0 \qquad \forall i \in \{1 \ldots n\}$$

$$I = \begin{pmatrix}1 & \mathbf{0}\\0 & \mathbf{1}\end{pmatrix} M = \begin{pmatrix}\mathbf{2} & 1\\\mathbf{1} & 3\end{pmatrix} q = \begin{pmatrix}-1\\-1\end{pmatrix}$$

Geometric Interpretation



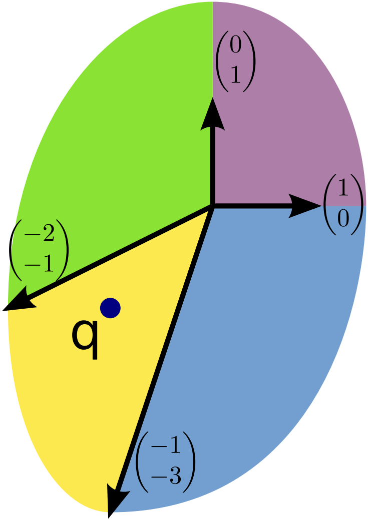
$$q = Iw - Mz$$

$$w, z \geq 0$$

$$z_i w_i = 0 \qquad \forall i \in \{1 \ldots n\}$$

$$I = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} \; M = \begin{pmatrix}\mathbf{2} & \mathbf{1}\\\mathbf{1} & \mathbf{3}\end{pmatrix} \; q = \begin{pmatrix}-1\\-1\end{pmatrix}$$

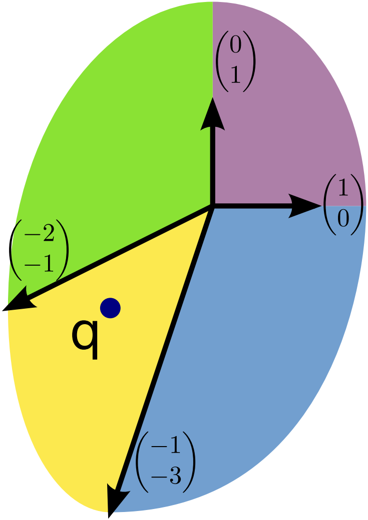
Geometric Interpretation



$$q = Iw - Mz$$
$$w, z \geq 0$$
$$z_i w_i = 0 \qquad \forall i \in \{1 \ldots n\}$$

$$I = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} M = \begin{pmatrix}2 & 1\\1 & 3\end{pmatrix} q = \begin{pmatrix}-1\\-1\end{pmatrix}$$

Geometric Interpretation

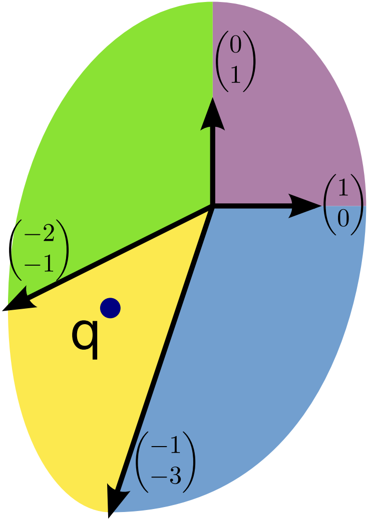


$$q = Iw - Mz$$
$$w, z \geq 0$$
$$z_i w_i = 0 \qquad \forall i \in \{1 \ldots n\}$$

Definition

A matrix is a P-matrix iff all its principal minors are positive

Geometric Interpretation

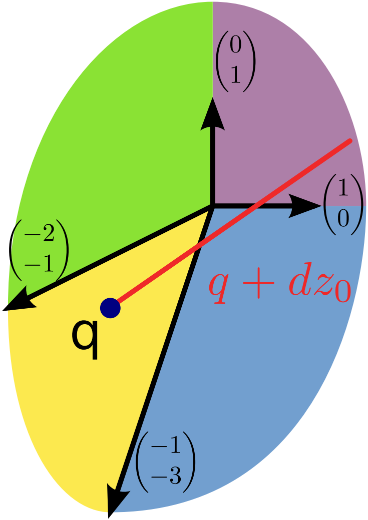


$$q = Iw - Mz$$
$$w, z \geq 0$$
$$z_i w_i = 0 \qquad \forall i \in \{1 \ldots n\}$$

Definition
A matrix is a P-matrix iff all its principal minors are positive

Theorem
M is a P-matrix iff the LCP has a unique solution for every q

Geometric Interpretation



$$
\begin{aligned}
q + dz_0 &= Iw - Mz \\
w, z &\geq 0 \\
z_i w_i &= 0 \qquad \forall i \in \{1 \dots n\}
\end{aligned}
$$

Discounted Games

Solving a discounted game can be reduced to P-matrix LCP (Jurdziński and Savani 2008)

Discounted Games

Solving a discounted game can be reduced to P-matrix LCP (Jurdziński and Savani 2008)

- M is determined by the graph structure
  - A complementary cone corresponds to a positional strategy
- q is determined by both the graph and the weights

Discounted Games

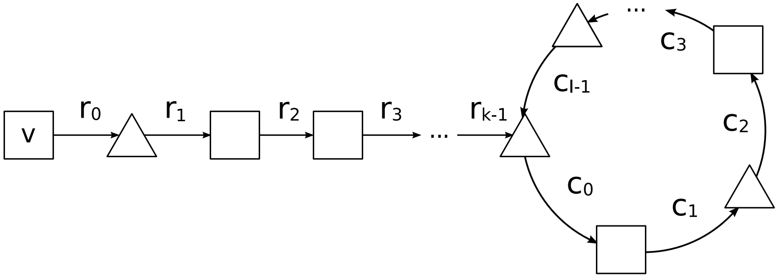
Solving a discounted game can be reduced to P-matrix LCP (Jurdziński and Savani 2008)

- M is determined by the graph structure
  - A complementary cone corresponds to a positional strategy
- q is determined by both the graph and the weights

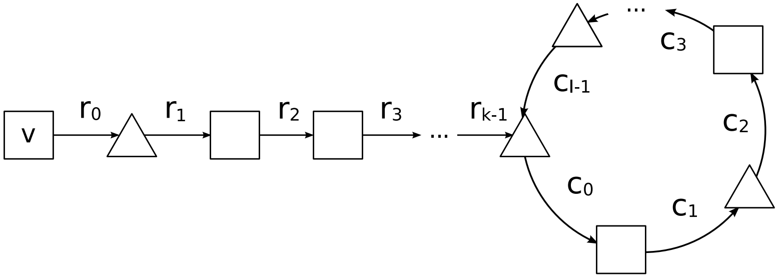
Our contribution is to

- Give a description of Lemke's algorithm for discounted games
- Show that it can take an exponential amount of time

Computing the Value of a Vertex



Computing the Value of a Vertex

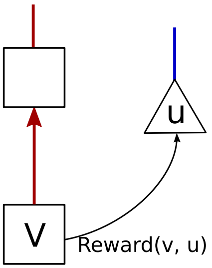


$$\text{Value}(v) = \sum_{i=0}^{k-1} \beta^i \text{r}_\text{i} + \sum_{i=0}^{l} \frac{\beta^{k+i}}{1-\beta^l} \text{c}_\text{i}$$

The Balance of a Vertex

For a Max vertex:

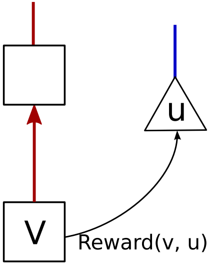
$$\text{Balance(v)} = \text{Value(v)} - (\text{Reward(v,u)} + \beta \times \text{Value(u)})$$



The Balance of a Vertex

For a Max vertex:

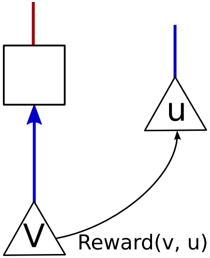
$$\text{Balance(v)} = \text{Value(v)} - (\text{Reward(v,u)} + \beta \times \text{Value(u)})$$



The balance compares the two successors

- Positive when the current successor is better
- Negative when the other successor is better

The Balance of a Vertex



For a Max vertex:

$$\text{Balance}(v) = \text{Value}(v) - (\text{Reward}(v,u) + \beta \times \text{Value}(u))$$

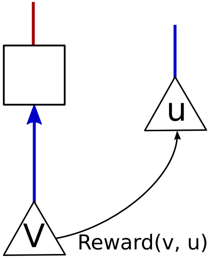
For a Min vertex:

$$\text{Balance}(v) = (\text{Reward}(v, u) + \beta \times \text{Value}(u)) - \text{Value}(v)$$

The balance compares the two successors

- Positive when the current successor is better
- Negative when the other successor is better

The Balance of a Vertex



For a Max vertex:

$$\text{Balance}(v) = \text{Value}(v) - (\text{Reward}(v,u) + \beta \times \text{Value}(u))$$

For a Min vertex:

$$\text{Balance}(v) = (\text{Reward}(v, u) + \beta \times \text{Value}(u)) - \text{Value}(v)$$

The balance compares the two successors

- Positive when the current successor is better
- Negative when the other successor is better

Theorem (Shapley 1953)

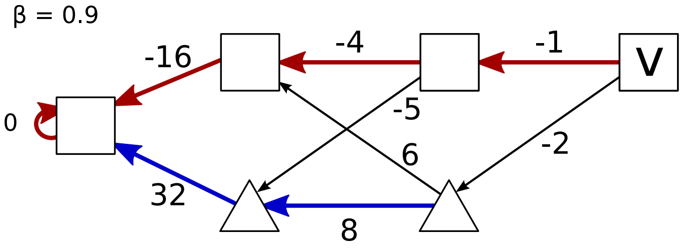
A pair of strategies is optimal if and only if no vertex has a negative balance

Lemke's Algorithm

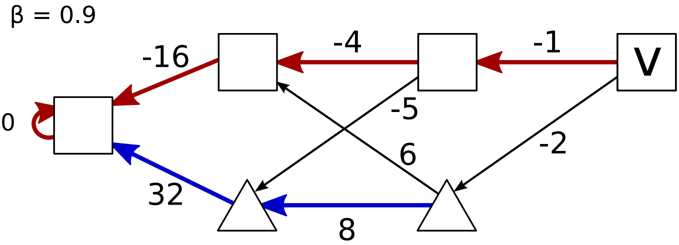
The idea

- Choose a pair of arbitrary strategies
- Modify the game to make the pair optimal (the trivial solution)
- Transform the modified game back to the original while ensuring that strategies are optimal

Lemke's Algorithm

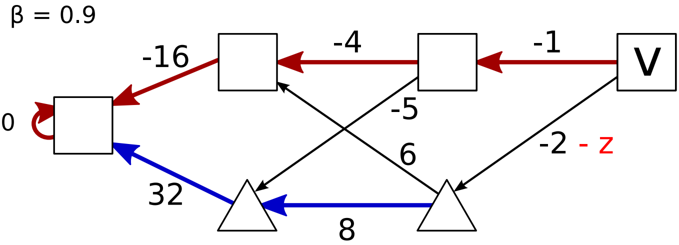


Lemke's Algorithm



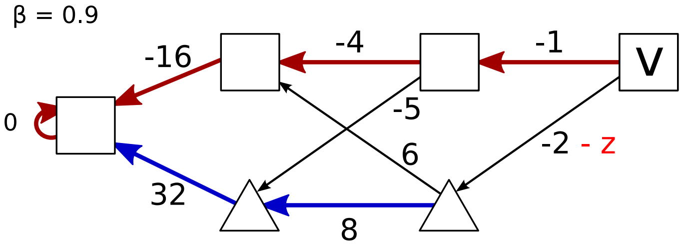
$$\begin{aligned}\text{Balance(v)} &= \text{Value(v)} - (\text{Reward(v,u)} + \beta \times \text{Value(u)}) \\ &= (-1 - \beta \times 4 - \beta^2 \times 16) - (-2 + \beta \times 8 + \beta^2 \times 32) \\ &= -48.68\end{aligned}$$

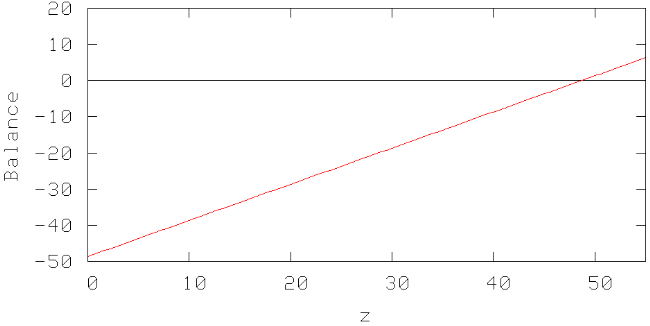
Lemke's Algorithm



$$
\begin{aligned}
\text{Balance(v)} &= \text{Value(v)} - (\text{Reward(v,u)} + \beta \times \text{Value(u)}) \\
&= (-1 - \beta \times 4 - \beta^2 \times 16) - (-2 - z + \beta \times 8 + \beta^2 \times 32) \\
&= -48.68
\end{aligned}
$$

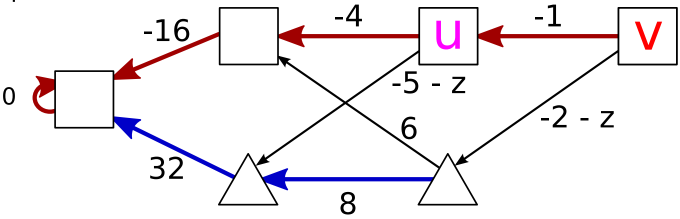
Lemke's Algorithm

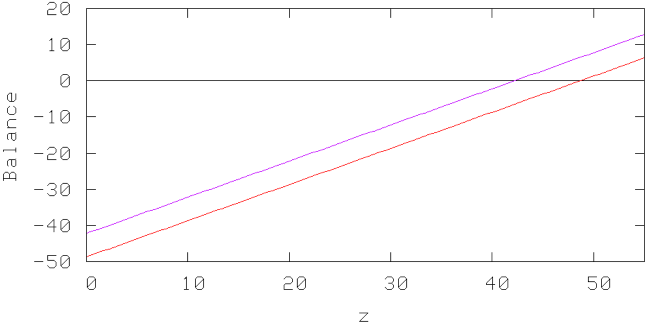




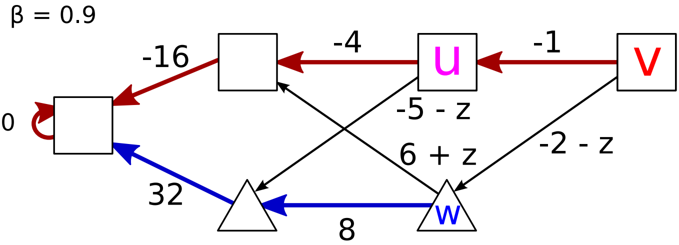
Lemke's Algorithm

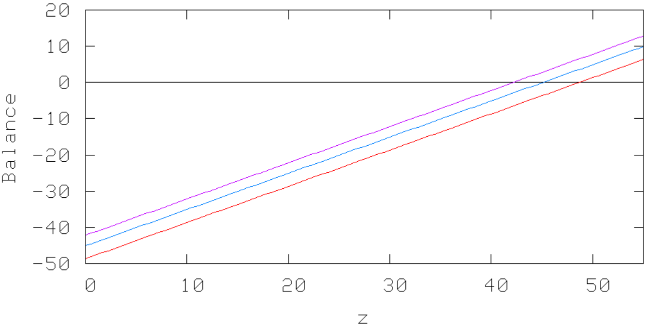
$\beta = 0.9$



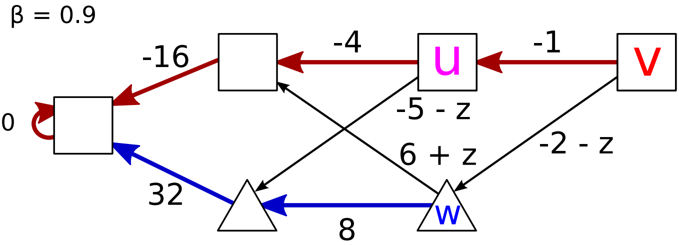


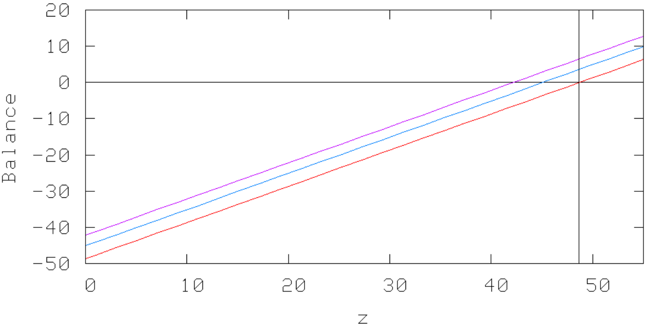
Lemke's Algorithm



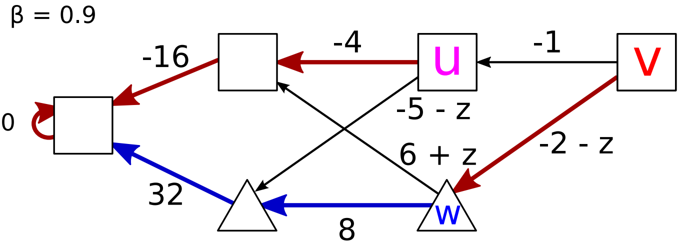


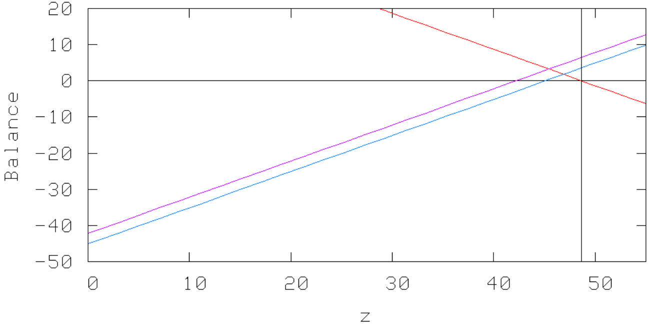
Lemke's Algorithm



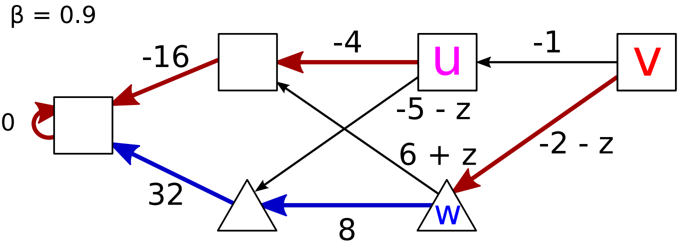


Lemke's Algorithm





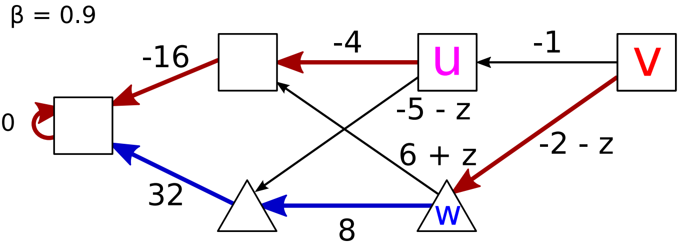
Lemke's Algorithm

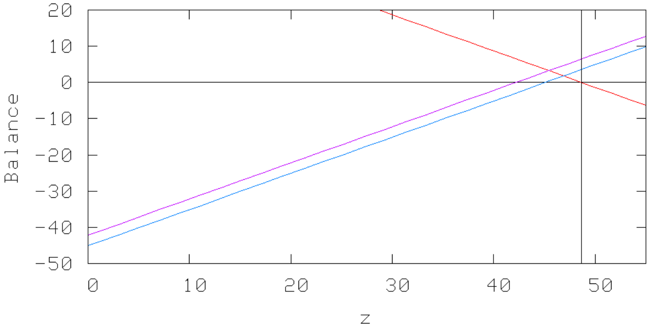


$$\text{Balance(v)} = \text{Value(v)} - (\text{Reward(v,u)} + \beta \times \text{Value(u)})$$

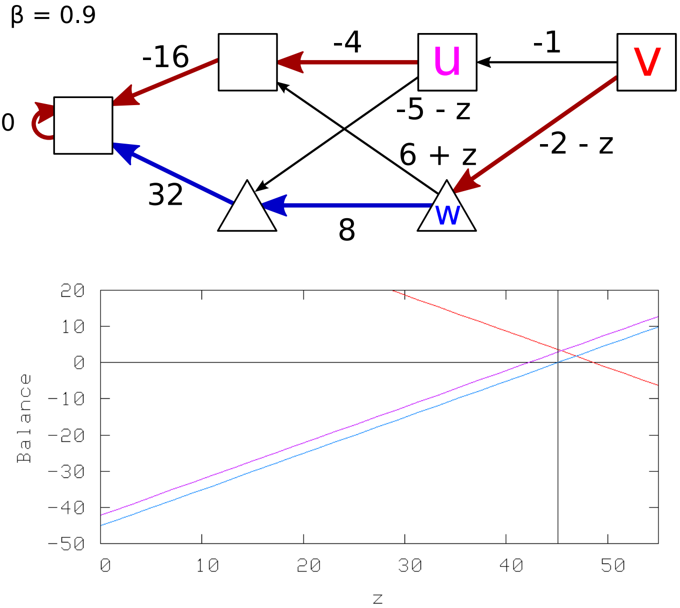
$$= (-2 - z + \beta \times 8 + \beta^2 \times 32) - (-1 - \beta \times 4 - \beta^2 \times 16)$$

Lemke's Algorithm

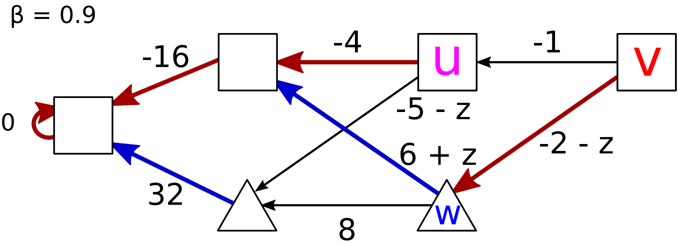


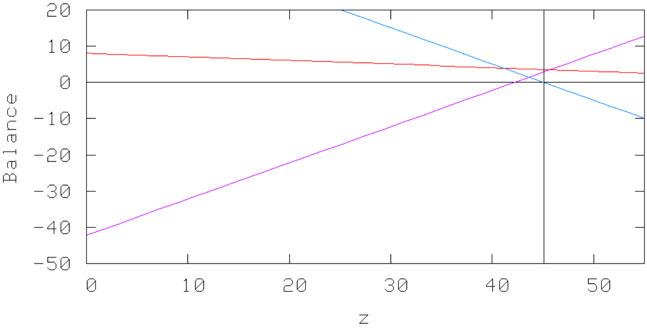


Lemke's Algorithm

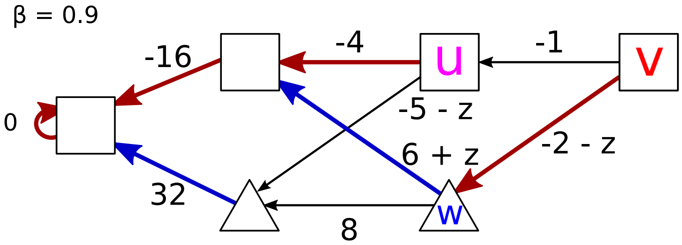


Lemke's Algorithm



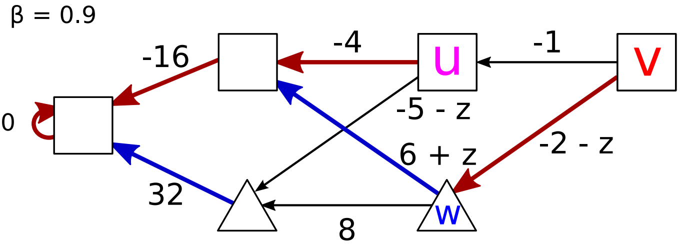


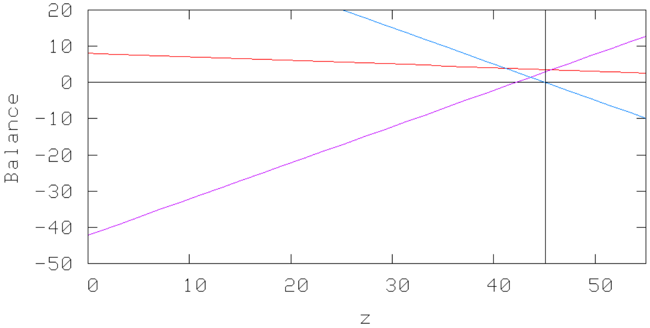
Lemke's Algorithm



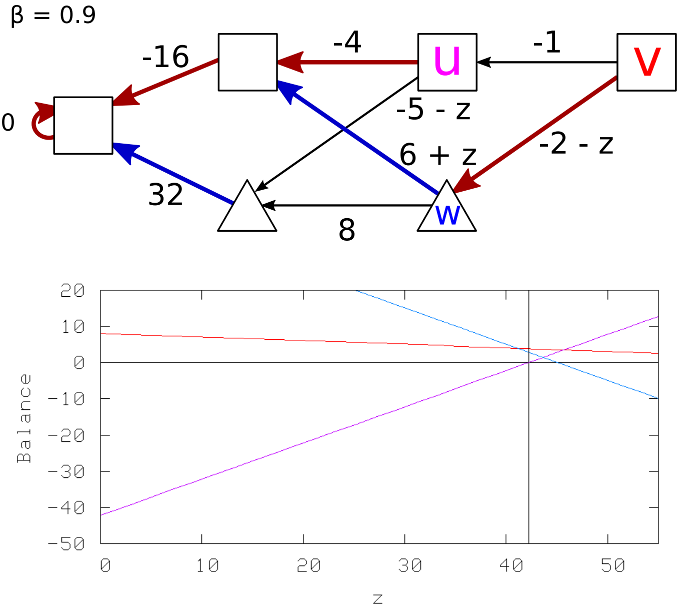
$$\text{Balance(v)} = \text{Value(v)} - (\text{Reward(v,u)} + \beta \times \text{Value(u)})$$
$$= (-2 - z + \beta \times (6+z) - \beta^2 \times 16)$$
$$- (-1 - \beta \times 4 - \beta^2 \times 16)$$

Lemke's Algorithm

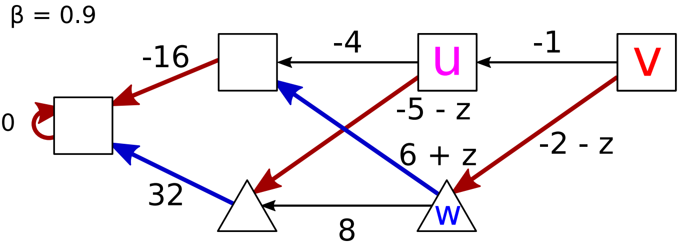


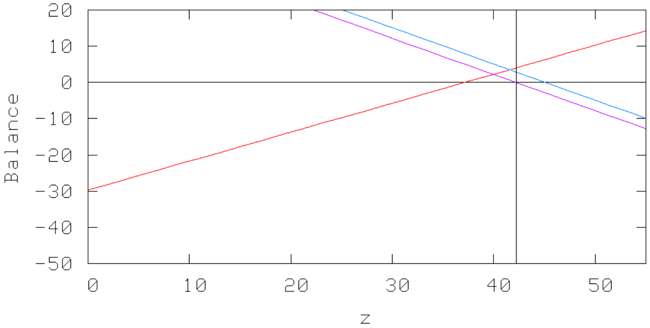


Lemke's Algorithm

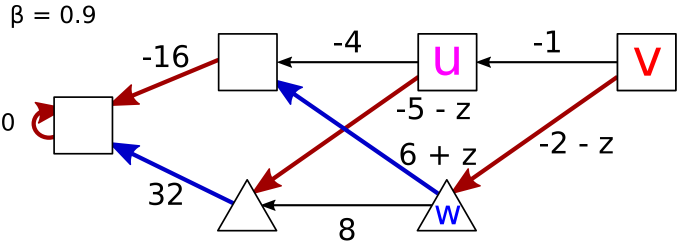


Lemke's Algorithm



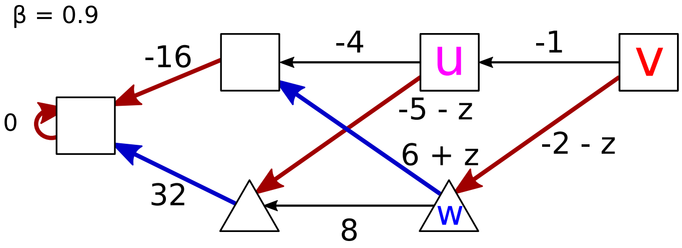


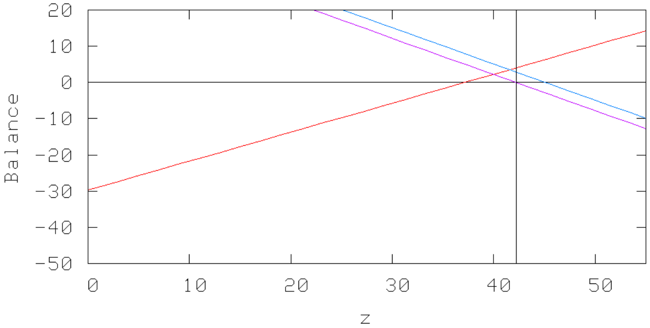
Lemke's Algorithm



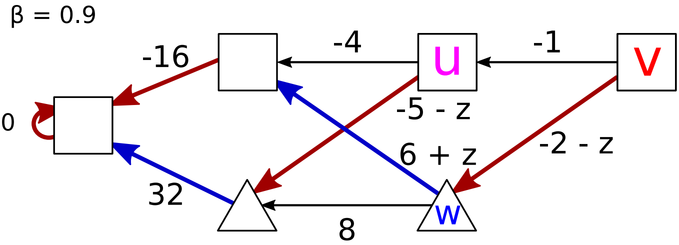
$$\text{Balance(v)} = \text{Value(v)} - (\text{Reward(v,u)} + \beta \times \text{Value(u)})$$
$$= (-2 - z + \beta \times (6+z) - \beta^2 \times 16)$$
$$-(-1 + \beta \times (5-z) + \beta^2 \times 32)$$

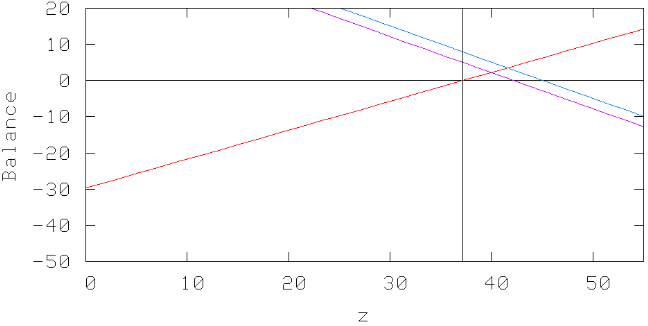
Lemke's Algorithm



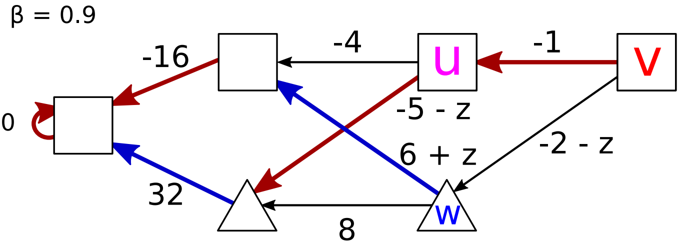


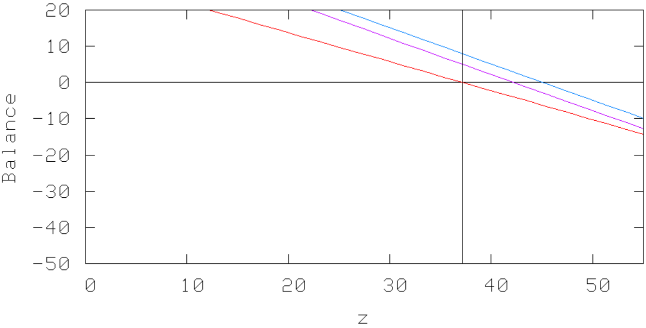
Lemke's Algorithm



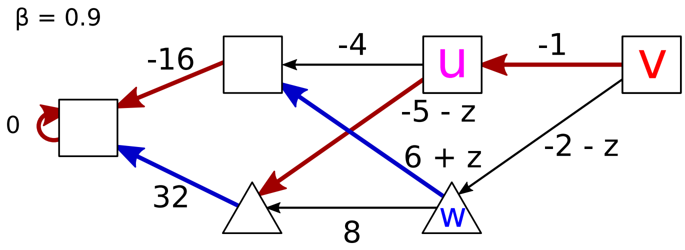


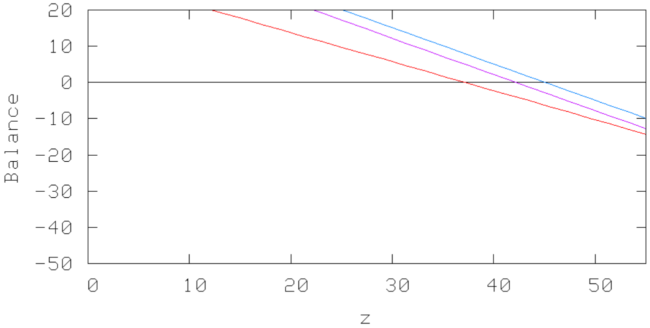
Lemke's Algorithm



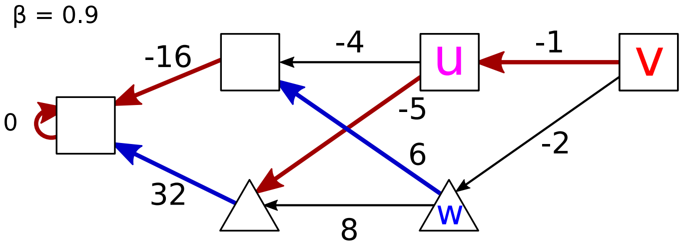


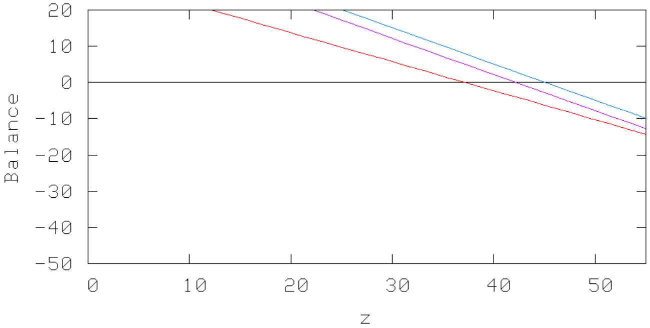
Lemke's Algorithm





Lemke's Algorithm





Lemke's Algorithm

In general, for each vertex we:

- Find the rate of change of the balance with respect to z
- Determine how far z can be decreased before the balance is 0

Lemke's Algorithm

In general, for each vertex we:

- Find the rate of change of the balance with respect to z
- Determine how far z can be decreased before the balance is 0

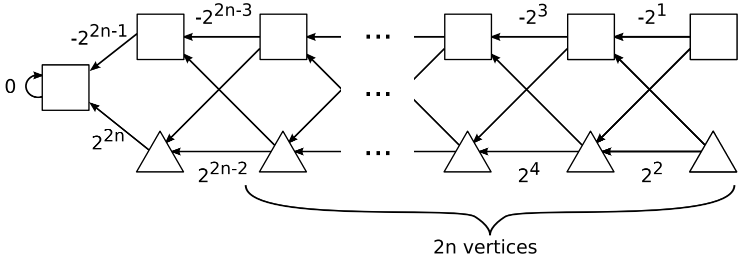
The minimum over all vertices shows us how far z can be decreased before we must change the strategy

Discounted Games

Our contribution is to

- Give a description of Lemke's algorithm for discounted games
- Show that it can take an exponential amount of time

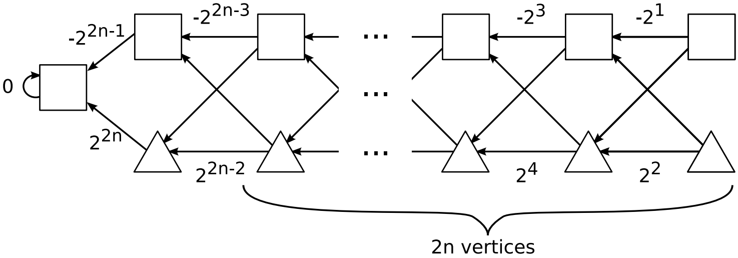
An Exponential-Time Example



It has been used before to show lower bounds for:

- Strategy improvement for 1 player simple-stochastic games (Melekopoglou and Condon 1994)
- Strategy improvement for mean-payoff games (Björklund and Vorobyov 2007)

An Exponential-Time Example

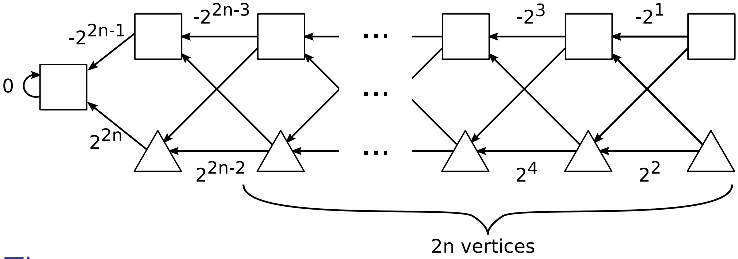


It has been used before to show lower bounds for:

- Strategy improvement for 1 player simple-stochastic games (Melekopoglou and Condon 1994)
- Strategy improvement for mean-payoff games (Björklund and Vorobyov 2007)

It works because the underlying combinatorial structure is a Klee-Minty cube.

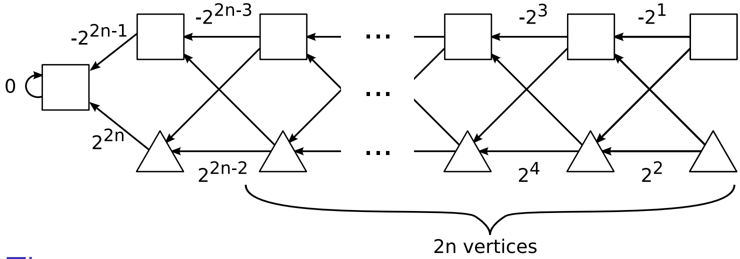
An Exponential-Time Example



Theorem

When β is close to 1 Lemke's algorithm will take an exponential number of steps

An Exponential-Time Example



Theorem

When β is close to 1 Lemke's algorithm will take an exponential number of steps

Corollary

Lemke's algorithm is exponential for parity games

Conclusion

We have seen:

- A new algorithm for discounted games
- An example upon which it takes an exponential number of steps

Conclusion

We have seen:

- A new algorithm for discounted games
- An example upon which it takes an exponential number of steps

We have also done this for the Cottle-Dantzig algorithm.

Future work

The lower bounds depend on a specific starting strategy and a specific choice of covering vector

What happens with:

- a random initial strategy?
- a random covering vector?

Future work

The lower bounds depend on a specific starting strategy and a specific choice of covering vector

What happens with:

- a random initial strategy?
- a random covering vector?

Theorem (Adler & Meggido 1985)

Lemke is quadratic on random linear programs

Can we extend this to discounted games?