# Solution of Peter Winkler's Pizza Problem

Viola Mészáros (Charles University)

Joint work with Josef Cibulka, Jan Kynčl, Rudolf Stolař and Pavel Valtr

# A Problem of Peter Winkler

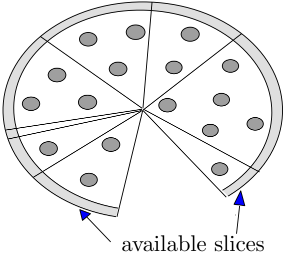


Figure: Bob and Alice are sharing a pizza

## A Problem of Peter Winkler

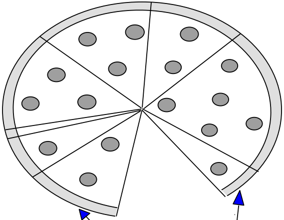


Figure: Bob and Alice are sharing a pizza

How much can Alice gain?

## Easy observations

- Bob can obtain half of the pizza by cutting the pizza into an even number of slices of equal size.

## Easy observations

- Bob can obtain half of the pizza by cutting the pizza into an even number of slices of equal size.
- If the number of slices is even, Alice has a strategy to gain at least half of the pizza.

# Shifts and jumps

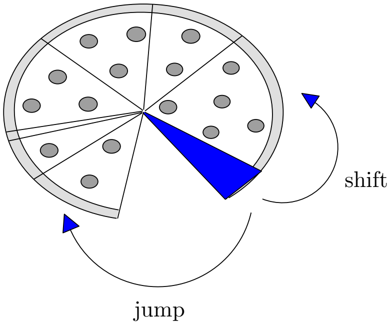


Figure: The two possible moves

# Shifts and jumps

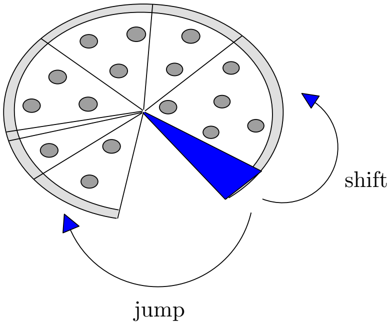


Figure: The two possible moves

If some strategy of a player allows the player to make at most $j$ jumps, then it is a *$j$-jump strategy*.

## Definitions

- The pizza may be represented by a circular sequence $P = p_0 p_1 \dots p_{n-1}$ and by *the weights* $|p_i| \geq 0$ for $(i = 0, 1, \dots, n-1)$.

## Definitions

- The pizza may be represented by a circular sequence $P = p_0 p_1 \dots p_{n-1}$ and by *the weights* $|p_i| \geq 0$ for $(i = 0, 1, \dots, n-1)$.
- *The weight of P* is defined by $|P| := \sum_{i=0}^{n-1} |p_i|$.

## Definitions

- The pizza may be represented by a circular sequence $P = p_0 p_1 \ldots p_{n-1}$ and by *the weights* $|p_i| \geq 0$ for $(i = 0, 1, \ldots, n-1)$.
- *The weight of P* is defined by $|P| := \sum_{i=0}^{n-1} |p_i|$.
- A player has a strategy with *gain g* if that strategy guarantees the player a subset of slices with sum of weights at least $g$.

## Restricting jumps

**Claim**

*Alice has a zero-jump strategy with gain $|P|/3$ and the constant $1/3$ is the best possible.*

## Restricting jumps

### Claim

*Alice has a zero-jump strategy with gain $|P|/3$ and the constant $1/3$ is the best possible.*

### Theorem

*Alice has a one-jump strategy with gain $7|P|/16$ and the constant $7/16$ is the best possible.*

## Restricting jumps

**Claim**

*Alice has a zero-jump strategy with gain $|P|/3$ and the constant $1/3$ is the best possible.*

**Theorem**

*Alice has a one-jump strategy with gain $7|P|/16$ and the constant $7/16$ is the best possible.*

Our main result:

**Theorem**

*For any $P$, Alice has a two-jump strategy with gain $4|P|/9$ and the constant $4/9$ is the best possible.*

## Characteristic cycle

If the number of slices is odd, instead of the circular sequence $P = p_0 p_1 \dots p_{n-1}$ consider *the characteristic cycle* defined as $V = v_0 v_1 \dots v_{n-1} = p_0 p_2 \dots p_{n-1} p_1 p_3 \dots p_{n-2}$.

## Characteristic cycle

If the number of slices is odd, instead of the circular sequence $P = p_0 p_1 \dots p_{n-1}$ consider *the characteristic cycle* defined as $V = v_0 v_1 \dots v_{n-1} = p_0 p_2 \dots p_{n-1} p_1 p_3 \dots p_{n-2}$.

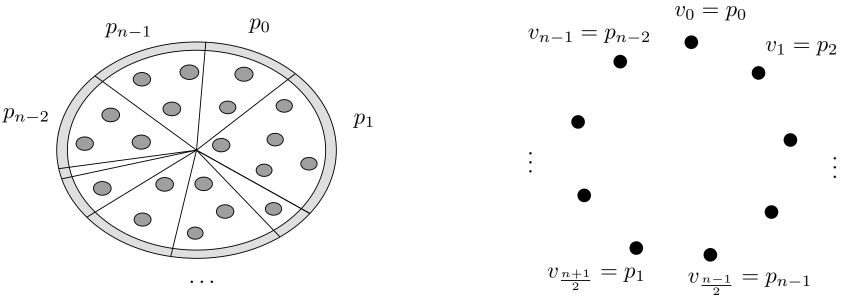


Figure: A cutting of a pizza and the corresponding characteristic cycle.

# A game

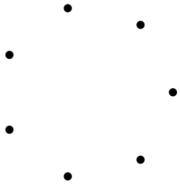

Figure: Turns: $A_1, B_2, A_3, \ldots,$ jumps: $B_4$ and $A_5$.

## A game

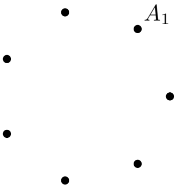


Figure: Turns: $A_1, B_2, A_3, \ldots,$ jumps: $B_4$ and $A_5$.

# A game

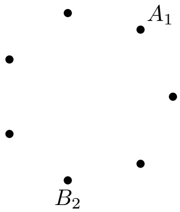


Figure: Turns: $A_1, B_2, A_3, \ldots,$ jumps: $B_4$ and $A_5$.

## A game

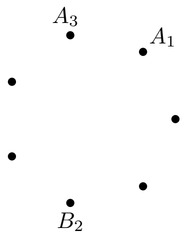


Figure: Turns: $A_1, B_2, A_3, \ldots$, jumps: $B_4$ and $A_5$.

# A game

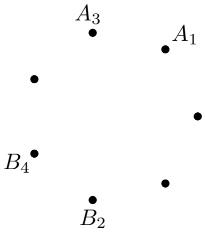


Figure: Turns: $A_1, B_2, A_3, \ldots,$ jumps: $B_4$ and $A_5$.

# A game

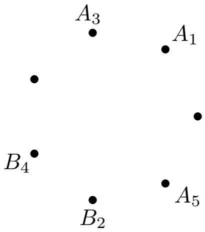


Figure: Turns: $A_1, B_2, A_3, \ldots,$ jumps: $B_4$ and $A_5$.

# A game

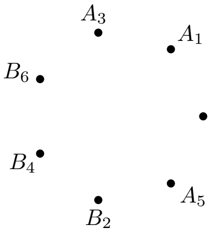


Figure: Turns: $A_1, B_2, A_3, \ldots$, jumps: $B_4$ and $A_5$.

## A game

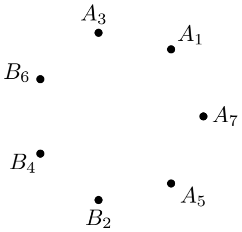


Figure: Turns: $A_1, B_2, A_3, \ldots$, jumps: $B_4$ and $A_5$.

## Further definitions

- An *arc* is a sequence of at most $n-1$ consecutive elements of $V$.

## Further definitions

- An *arc* is a sequence of at most $n-1$ consecutive elements of $V$.
- For an arc $X = v_i v_{i+1} \dots v_{i+k-1}$, its *length* is $l(X) := k$.

## Further definitions

- An *arc* is a sequence of at most $n-1$ consecutive elements of $V$.
- For an arc $X = v_i v_{i+1} \dots v_{i+k-1}$, its *length* is $l(X) := k$.
- *The weight* of $X$ is $|X| := \sum_{j=i}^{i+k-1} |v_j|$.

## Further definitions

- An *arc* is a sequence of at most $n-1$ consecutive elements of $V$.
- For an arc $X = v_i v_{i+1} \dots v_{i+k-1}$, its *length* is $l(X) := k$.
- *The weight* of $X$ is $|X| := \sum_{j=i}^{i+k-1} |v_j|$.
- An arc of length $(n+1)/2$ is called a *half-circle*.

## Further definitions

- An *arc* is a sequence of at most $n-1$ consecutive elements of $V$.
- For an arc $X = v_i v_{i+1} \dots v_{i+k-1}$, its *length* is $l(X) := k$.
- *The weight* of $X$ is $|X| := \sum_{j=i}^{i+k-1} |v_j|$.
- An arc of length $(n+1)/2$ is called a *half-circle*.
- For each $v$ in $V$ *the potential of v* is the minimum of the weights of half-circles covering $v$.

## Zero-jump strategy

- Lower bound: There exists $v$ on $V$ with potential at least $1/3$.

# Zero-jump strategy

- Lower bound: There exists $v$ on $V$ with potential at least $1/3$.

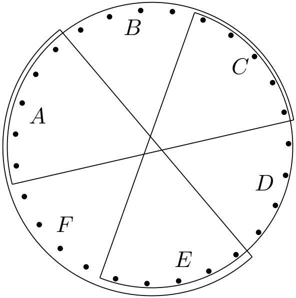


Figure: A covering triple of half-circles.

## Zero-jump strategy

- Lower bound: There exists $v$ on $V$ with potential at least $1/3$.

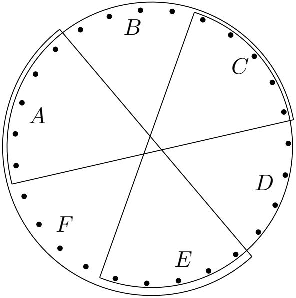


Figure: A covering triple of half-circles.

- Upper bound: Consider the cutting $V = 100100100$.

## One-jump strategy

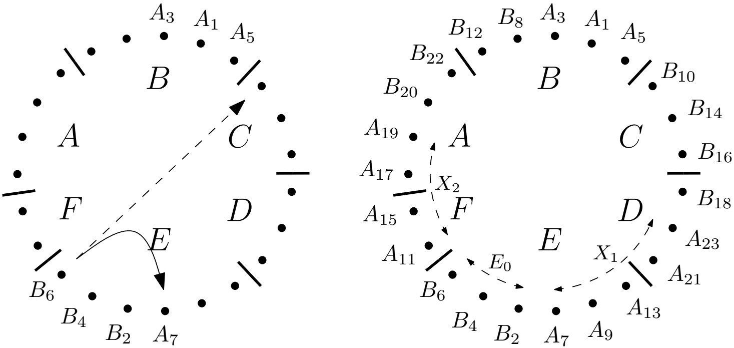


Figure: One-jump strategy: Alice chooses a jump rather than a shift (left) and makes no more jumps afterwards (right).

# Two-jump strategy

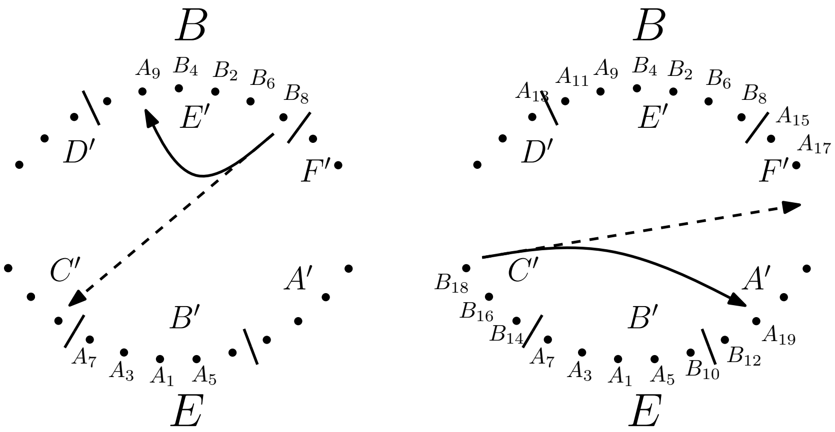


Figure: We define two phases of the game. During the first phase Alice makes one jump (left). She makes another jump as the first turn of the second phase (right).

## Analysis of Alice's gain

For $n \geq 1$, let $g(n)$ be the maximum $g \in [0,1]$ such that for any cutting of the pizza into $n$ slices, Alice has a strategy with gain $g|P|$.

## Analysis of Alice's gain

For $n \geq 1$, let $g(n)$ be the maximum $g \in [0,1]$ such that for any cutting of the pizza into $n$ slices, Alice has a strategy with gain $g|P|$.

**Theorem**

*Let $n \geq 1$. Then*

$$g(n) = \begin{cases} 1 & \text{if } n = 1, \\ 4/9 & \text{if } n \in \{15, 17, 19, 21, \dots\}, \\ 1/2 & \text{otherwise.} \end{cases}$$

## Analysis of Alice's gain

For $n \geq 1$, let $g(n)$ be the maximum $g \in [0,1]$ such that for any cutting of the pizza into $n$ slices, Alice has a strategy with gain $g|P|$.

**Theorem**

*Let $n \geq 1$. Then*

$$g(n) = \begin{cases} 1 & \text{if } n = 1, \\ 4/9 & \text{if } n \in \{15, 17, 19, 21, \dots\}, \\ 1/2 & \text{otherwise.} \end{cases}$$

*Alice uses a zero-jump strategy when $n$ is even or $n \leq 7$, a one-jump strategy for $n \in \{9, 11, 13\}$, and a two-jump strategy for $n \in \{15, 17, 19, 21, \dots\}$.*

## Some more results

### Theorem

*For any* $\omega \in [0,1]$*, Bob has a one-jump strategy with gain* $5|P|/9$ *if he cuts the pizza into* 15 *slices as follows:* $P_\omega = 0010100(1+\omega)0(2-\omega)00202$*. These cuttings describe, up to scaling, rotating and flipping the pizza upside-down, all the pizza cuttings into* 15 *slices for which Bob has a strategy with gain* $5|P|/9$*.*

## Some more results

**Theorem**

*For any* $\omega \in [0,1]$*, Bob has a one-jump strategy with gain* $5|P|/9$ *if he cuts the pizza into* 15 *slices as follows:* $P_\omega = 0010100(1+\omega)0(2-\omega)00202$. *These cuttings describe, up to scaling, rotating and flipping the pizza upside-down, all the pizza cuttings into* 15 *slices for which Bob has a strategy with gain* $5|P|/9$.

**Theorem**

*Up to scaling, rotating and flipping the pizza upside-down, there is a unique pizza cutting into* 21 *slices of at most two different sizes for which Bob has a strategy with gain* $5|P|/9$. *The cutting is* 001010010101001010101.

## Algorithms

### Theorem

*There is an algorithm that, given a cutting of the pizza with $n$ slices, performs a precomputation in time $O(n)$. Then, during the game, the algorithm decides each of Alice's turns in time $O(1)$ in such a way that Alice makes at most two jumps and her gain is at least $g(n)|P|$.*

## Algorithms

### Theorem

*There is an algorithm that, given a cutting of the pizza with $n$ slices, performs a precomputation in time $O(n)$. Then, during the game, the algorithm decides each of Alice's turns in time $O(1)$ in such a way that Alice makes at most two jumps and her gain is at least $g(n)|P|$.*

### Claim

*There is an algorithm that, given a cutting of the pizza with $n$ slices, computes an optimal strategy for each of the two players in time $O(n^2)$. The algorithm stores an optimal turn of the player on turn for all the $n^2 - n + 2$ possible positions of the game.*

## Open problem

### Problem

*Is there an algorithm that uses $o(n^2)$ time for some precomputations and then computes each optimal turn in constant time?*