# Finding small domatic partitions in graphs with large domatic number

Oscar Levin

University of Northern Colorado

ASL North American Annual Meeting
May 20, 2014

Joint work with Matthew Jura and Tyler Markkanen

# Background and Motivation

- The chromatic number $\chi(G)$ of a graph is the size of the smallest partition of vertices into independent sets.
- The domatic number $d(G)$ of a graph is the size of the largest partition of vertices into dominating sets.
- A set $D$ of vertices is dominating if every vertex not in $D$ is adjacent to a vertex in $D$.

# Background and Motivation

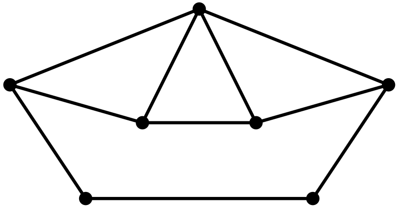
- The chromatic number $\chi(G)$ of a graph is the size of the smallest partition of vertices into independent sets.
- The domatic number $d(G)$ of a graph is the size of the largest partition of vertices into dominating sets.
- A set $D$ of vertices is dominating if every vertex not in $D$ is adjacent to a vertex in $D$.

# Background and Motivation

- The chromatic number $\chi(G)$ of a graph is the size of the smallest partition of vertices into independent sets.
- The domatic number $d(G)$ of a graph is the size of the largest partition of vertices into dominating sets.
- A set $D$ of vertices is dominating if every vertex not in $D$ is adjacent to a vertex in $D$.

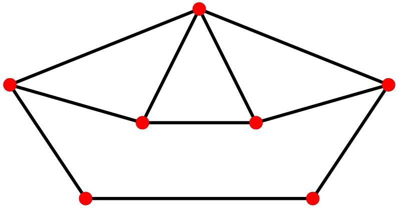
# Background and Motivation

- The chromatic number $\chi(G)$ of a graph is the size of the smallest partition of vertices into independent sets.
- The domatic number $d(G)$ of a graph is the size of the largest partition of vertices into dominating sets.
- A set $D$ of vertices is dominating if every vertex not in $D$ is adjacent to a vertex in $D$.



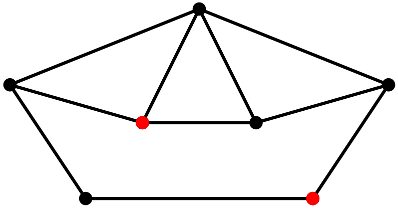
# Background and Motivation

- The chromatic number $\chi(G)$ of a graph is the size of the smallest partition of vertices into independent sets.
- The domatic number $d(G)$ of a graph is the size of the largest partition of vertices into dominating sets.
- A set $D$ of vertices is dominating if every vertex not in $D$ is adjacent to a vertex in $D$.



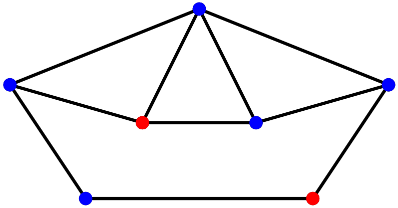
# Background and Motivation

- The chromatic number $\chi(G)$ of a graph is the size of the smallest partition of vertices into independent sets.
- The domatic number $d(G)$ of a graph is the size of the largest partition of vertices into dominating sets.
- A set $D$ of vertices is dominating if every vertex not in $D$ is adjacent to a vertex in $D$.



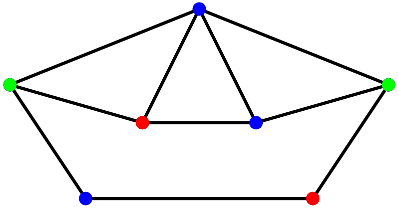
# Background and Motivation

- The chromatic number $\chi(G)$ of a graph is the size of the smallest partition of vertices into independent sets.
- The domatic number $d(G)$ of a graph is the size of the largest partition of vertices into dominating sets.
- A set $D$ of vertices is dominating if every vertex not in $D$ is adjacent to a vertex in $D$.



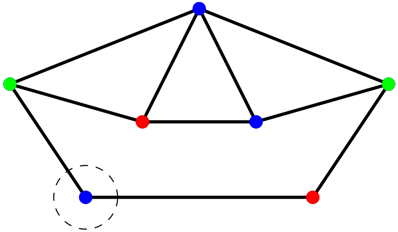
# Background and Motivation

- The chromatic number $\chi(G)$ of a graph is the size of the smallest partition of vertices into independent sets.
- The domatic number $d(G)$ of a graph is the size of the largest partition of vertices into dominating sets.
- A set $D$ of vertices is dominating if every vertex not in $D$ is adjacent to a vertex in $D$.



# Background and Motivation

- The chromatic number $\chi(G)$ of a graph is the size of the smallest partition of vertices into independent sets.
- The domatic number $d(G)$ of a graph is the size of the largest partition of vertices into dominating sets.
- A set $D$ of vertices is dominating if every vertex not in $D$ is adjacent to a vertex in $D$.



# Main Question

### Question

*Can you always find a domatic $n$-partition of a graph with domatic number $n$?*

*If not, is it be easier to find smaller domatic partitions?*

### Question (better)

*Given a computable graph $G$ with domatic number $n$, what is the size of the largest computable domatic partition of $G$?*

*In other words, what is $d^c(G)$, the computable domatic number?*

# Main Question

## Question

*Can you always find a domatic $n$-partition of a graph with domatic number $n$?*

*If not, is it be easier to find smaller domatic partitions?*

## Question (better)

*Given a computable graph $G$ with domatic number $n$, what is the size of the largest computable domatic partition of $G$?*

*In other words, what is $d^c(G)$, the computable domatic number?*

# If $d(G) \geq 2$ then $d^c(G) \geq 2$.

Suppose $G$ has a domatic 2-partition (so no isolated vertices).

There is an algorithm which produces a domatic 2-partition.

Vertices: $\{v_0, v_1, v_2, \ldots\}$

Put $v_0 \in A$.

Put $v_n \in B$ iff there is an adjacent vertex $v_k \in A$ (with $k < n$)

$A$ is a dominating set: if $v_n \notin A$ then ...

$B$ is a dominating set: if $v_n \notin B$ then ...

# If $d(G) \geq 2$ then $d^c(G) \geq 2$.

Suppose $G$ has a domatic 2-partition (so no isolated vertices).
There is an algorithm which produces a domatic 2-partition.

Vertices: $\{v_0, v_1, v_2, \ldots\}$

Put $v_0 \in A$.

Put $v_n \in B$ iff there is an adjacent vertex $v_k \in A$ (with $k < n$)

$A$ is a dominating set: if $v_n \notin A$ then …

$B$ is a dominating set: if $v_n \notin B$ then …

# If $d(G) \geq 2$ then $d^c(G) \geq 2$.

Suppose $G$ has a domatic 2-partition (so no isolated vertices).
There is an algorithm which produces a domatic 2-partition.

Vertices: $\{v_0, v_1, v_2, \ldots\}$

Put $v_0 \in A$.

Put $v_n \in B$ iff there is an adjacent vertex $v_k \in A$ (with $k < n$)

$A$ is a dominating set: if $v_n \notin A$ then ...

$B$ is a dominating set: if $v_n \notin B$ then ...

# If $d(G) \geq 2$ then $d^c(G) \geq 2$.

Suppose $G$ has a domatic 2-partition (so no isolated vertices).
There is an algorithm which produces a domatic 2-partition.

Vertices: $\{v_0, v_1, v_2, \ldots\}$

Put $v_0 \in A$.

Put $v_n \in B$ iff there is an adjacent vertex $v_k \in A$ (with $k < n$)

$A$ is a dominating set: if $v_n \notin A$ then ...

$B$ is a dominating set: if $v_n \notin B$ then ...

# If $d(G) \geq 2$ then $d^c(G) \geq 2$.

Suppose $G$ has a domatic 2-partition (so no isolated vertices).
There is an algorithm which produces a domatic 2-partition.

Vertices: $\{v_0, v_1, v_2, \ldots\}$

Put $v_0 \in A$.

Put $v_n \in B$ iff there is an adjacent vertex $v_k \in A$ (with $k < n$)

$A$ is a dominating set: if $v_n \notin A$ then ...

$B$ is a dominating set: if $v_n \notin B$ then ...

# If $d(G) \geq 2$ then $d^c(G) \geq 2$.

Suppose $G$ has a domatic 2-partition (so no isolated vertices). There is an algorithm which produces a domatic 2-partition.

Vertices: $\{v_0, v_1, v_2, \ldots\}$

Put $v_0 \in A$.

Put $v_n \in B$ iff there is an adjacent vertex $v_k \in A$ (with $k < n$)

$A$ is a dominating set: if $v_n \notin A$ then ...

$B$ is a dominating set: if $v_n \notin B$ then ...

# If $d(G) \geq 2$ then $d^c(G) \geq 2$.

Suppose $G$ has a domatic 2-partition (so no isolated vertices).
There is an algorithm which produces a domatic 2-partition.

Vertices: $\{v_0, v_1, v_2, \ldots\}$

Put $v_0 \in A$.

Put $v_n \in B$ iff there is an adjacent vertex $v_k \in A$ (with $k < n$)

$A$ is a dominating set: if $v_n \notin A$ then ...

$B$ is a dominating set: if $v_n \notin B$ then ...

# What if $d(G) = 3$?

**Proposition**

*There is a computable graph $G$ with $d(G) = 3$ but $d^c(G) = 2$.*

To prove this, we diagonalize against all computable functions.

What until $\varphi_e$ partitions its copy of $K_4$. If it looks like a domatic 3-partition, spring the trap.

# What if $d(G) = 3$?

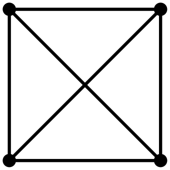
**Proposition**

*There is a computable graph $G$ with $d(G) = 3$ but $d^c(G) = 2$.*

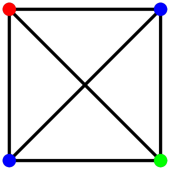
To prove this, we diagonalize against all computable functions.

What until $\varphi_e$ partitions its copy of $K_4$. If it looks like a domatic 3-partition, spring the trap.

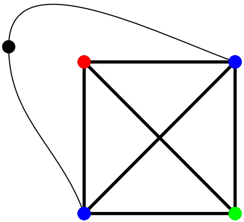
# The Trap



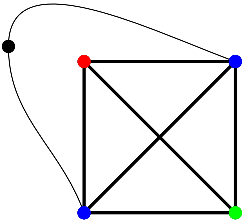
# The Trap



# The Trap

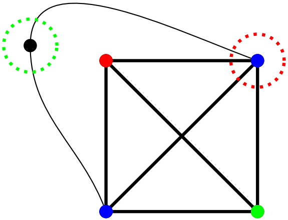


# The Trap



The sprung trap still has a 3-partition, but not the one $\varphi_e$ claims.

# The Trap



The sprung trap still has a 3-partition, but not the one $\varphi_e$ claims.

# ~~Worse~~ Better than that. . .

**Proposition**

*For any $n$, there is a computable graph $G$ with domatic number $d(G) = n$ but $d^c(G) = 2$.*

Use $K_{3(n-2)+1}$ as the trap to diagonalize against all possible computable domatic 3-partitions.

When $\varphi_e$ partitions its gadget, if it does so with three colors, there will be $n - 1$ vertices colored identically. Add a vertex adjacent to exactly those.

# ~~Worse~~ Better than that. . .

**Proposition**

*For any $n$, there is a computable graph $G$ with domatic number $d(G) = n$ but $d^c(G) = 2$.*

Use $K_{3(n-2)+1}$ as the trap to diagonalize against all possible computable domatic 3-partitions.

When $\varphi_e$ partitions its gadget, if it does so with three colors, there will be $n-1$ vertices colored identically. Add a vertex adjacent to exactly those.

# ~~Worse~~ Better than that...

**Proposition**

*For any $n$, there is a computable graph $G$ with domatic number $d(G) = n$ but $d^c(G) = 2$.*

Use $K_{3(n-2)+1}$ as the trap to diagonalize against all possible computable domatic 3-partitions.

When $\varphi_e$ partitions its gadget, if it does so with three colors, there will be $n - 1$ vertices colored identically. Add a vertex adjacent to exactly those.

# Stupid $\varphi_e$

Why does $\varphi_e$ partition its trap so soon?

Just because $G$ is computable, doesn't mean we can compute the degree of a given vertex!

But what if we could?

# Stupid $\varphi_e$

Why does $\varphi_e$ partition its trap so soon?

Just because $G$ is computable, doesn't mean we can compute the degree of a given vertex!

But what if we could?

# Stupid $\varphi_e$

Why does $\varphi_e$ partition its trap so soon?

Just because $G$ is computable, doesn't mean we can compute the degree of a given vertex!

But what if we could?

# Highly computable graphs

**Definition**

A graph is highly computable if it is computable and the degree function is computable.

Does this extra information help $\varphi_e$ compute a domatic partition?

**Proposition**

*There is a highly computable graph with domatic number 3 but computable domatic number 2.*

# Highly computable graphs

## Definition

A graph is highly computable if it is computable and the degree function is computable.

Does this extra information help $\varphi_e$ compute a domatic partition?

## Proposition

*There is a highly computable graph with domatic number 3 but computable domatic number 2.*

# The idea: remotely sprung traps

Wait for $\varphi_e$ to partition some fixed vertices. Then act.

Our action cannot change the degree of any vertex in the graph.

$\varphi_e$ might never partition its vertices, but we don't know that at any finite stage.

We must be able to force $\varphi_e$'s partition to be wrong, by modifying the graph arbitrarily far away from $\varphi_e$'s vertices.

# The idea: remotely sprung traps

Wait for $\varphi_e$ to partition some fixed vertices. Then act.

Our action cannot change the degree of any vertex in the graph.

$\varphi_e$ might never partition its vertices, but we don't know that at any finite stage.

We must be able to force $\varphi_e$'s partition to be wrong, by modifying the graph arbitrarily far away from $\varphi_e$'s vertices.

# The idea: remotely sprung traps

Wait for $\varphi_e$ to partition some fixed vertices. Then act.

Our action cannot change the degree of any vertex in the graph.

$\varphi_e$ might never partition its vertices, but we don't know that at any finite stage.

We must be able to force $\varphi_e$'s partition to be wrong, by modifying the graph arbitrarily far away from $\varphi_e$'s vertices.

# The idea: remotely sprung traps

Wait for $\varphi_e$ to partition some fixed vertices. Then act.

Our action cannot change the degree of any vertex in the graph.

$\varphi_e$ might never partition its vertices, but we don't know that at any finite stage.

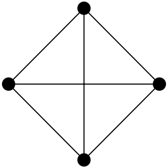
We must be able to force $\varphi_e$'s partition to be wrong, by modifying the graph arbitrarily far away from $\varphi_e$'s vertices.

# Highly intricate trap

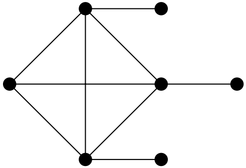
A path:

Every third vertex must be colored the same.

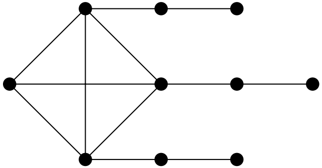
# Springing the trap



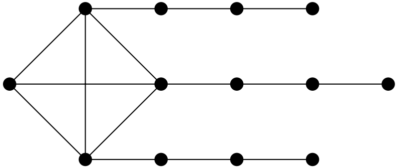
# Springing the trap



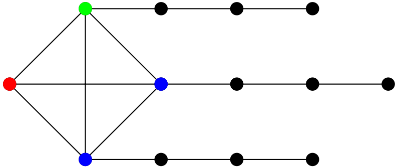
# Springing the trap



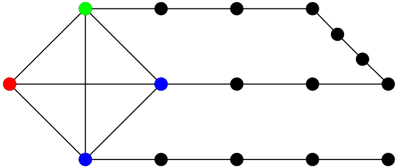
# Springing the trap



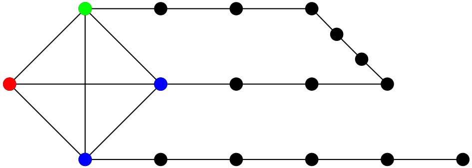
# Springing the trap



# Springing the trap



# Springing the trap



# If $d(G) = n$ then...

**Proposition**

*There is a highly computable graph $G$ with $d(G) = 4$ but $d^c(G) = 3$.*



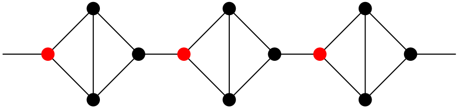
**Proposition**

*For all $n > 2$, there is a highly computable graph with $d(G) = n$ but $d^c(G) = n - 1$*

# If $d(G) = n$ then...

**Proposition**

*There is a highly computable graph $G$ with $d(G) = 4$ but $d^c(G) = 3$.*



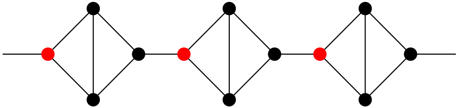
**Proposition**

*For all $n > 2$, there is a highly computable graph with $d(G) = n$ but $d^c(G) = n - 1$*

# If $d(G) = n$ then...

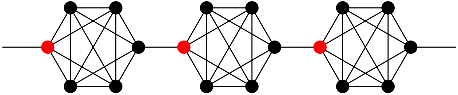
## Proposition

*There is a highly computable graph $G$ with $d(G) = 4$ but $d^c(G) = 3$.*



## Proposition

*For all $n > 2$, there is a highly computable graph with $d(G) = n$ but $d^c(G) = n - 1$*



# Can we do better?

Is it easier to find smaller domatic partitions in highly computable graphs?

**Conjecture**

*Any highly computable graph with domatic number $n$ has computable domatic number at least $f(n)$.*

Maybe $f(n) = n - 1$. Or $f(n) = (n + 1)/2$

# Can we do better?

Is it easier to find smaller domatic partitions in highly computable graphs?

**Conjecture**

*Any highly computable graph with domatic number $n$ has computable domatic number at least $f(n)$.*

Maybe $f(n) = n - 1$. Or $f(n) = (n + 1)/2$

# Can we do better?

Is it easier to find smaller domatic partitions in highly computable graphs?

**Conjecture**

*Any highly computable graph with domatic number $n$ has computable domatic number at least $f(n)$.*

Maybe $f(n) = n - 1$. Or $f(n) = (n + 1)/2$

# A partial result

**Proposition**

*Every highly computable 6-regular connected graph with domatic number 7 has a computable domatic 3-partition.*

(6-regular means the degree of every vertex is 6)

# Proof idea

Given a finite subgraph $H$, we can partition the vertices in and adjacent to $H$ so that every vertex in $H$ is $7$-dominated.

Start with a core $H_0$ and an $H_1$ surrounding $H_0$. Partition each (including neighbors) separately.

Now resolve the "double coloring." If $v$ is colored $c_1$ and $c_2$, color it:

- red if $c_1$ or $c_2$ is red, otherwise,
- blue if $c_1$ or $c_2$ is blue or green, otherwise
- yellow.

# Proof idea

Given a finite subgraph $H$, we can partition the vertices in and adjacent to $H$ so that every vertex in $H$ is $7$-dominated.

Start with a core $H_0$ and an $H_1$ surrounding $H_0$. Partition each (including neighbors) separately.

Now resolve the “double coloring.” If $v$ is colored $c_1$ and $c_2$, color it:

- red if $c_1$ or $c_2$ is red, otherwise,
- blue if $c_1$ or $c_2$ is blue or green, otherwise
- yellow.

# Proof idea

Given a finite subgraph $H$, we can partition the vertices in and adjacent to $H$ so that every vertex in $H$ is $7$-dominated.

Start with a core $H_0$ and an $H_1$ surrounding $H_0$. Partition each (including neighbors) separately.

Now resolve the “double coloring.” If $v$ is colored $c_1$ and $c_2$, color it:

- red if $c_1$ or $c_2$ is red, otherwise,
- blue if $c_1$ or $c_2$ is blue or green, otherwise
- yellow.

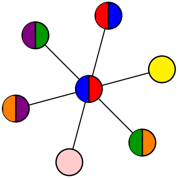
# Proof idea

Given a finite subgraph $H$, we can partition the vertices in and adjacent to $H$ so that every vertex in $H$ is 7-dominated.

Start with a core $H_0$ and an $H_1$ surrounding $H_0$. Partition each (including neighbors) separately.

Now resolve the “double coloring.” If $v$ is colored $c_1$ and $c_2$, color it:

- red if $c_1$ or $c_2$ is red, otherwise,
- blue if $c_1$ or $c_2$ is blue or green, otherwise
- yellow.



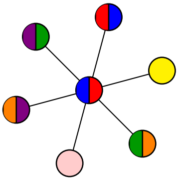
# Proof idea

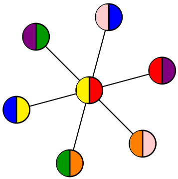
Given a finite subgraph $H$, we can partition the vertices in and adjacent to $H$ so that every vertex in $H$ is $7$-dominated.

Start with a core $H_0$ and an $H_1$ surrounding $H_0$. Partition each (including neighbors) separately.

Now resolve the "double coloring." If $v$ is colored $c_1$ and $c_2$, color it:

- red if $c_1$ or $c_2$ is red, otherwise,
- blue if $c_1$ or $c_2$ is blue or green, otherwise
- yellow.





# In general. . .

**Proposition**

*For any $n$ and any $k \geq n^2 - n + 1$, any highly computable, $k-1$-regular connected graph with domatic number $k$ has a computable domatic $n$-partition.*

Actually, the graphs need NOT be connected.

Getting rid of the regularity requirement appears to be much harder.

Proving that we cannot make $k$ smaller also appears really difficult.

# In general...

**Proposition**

*For any $n$ and any $k \geq n^2 - n + 1$, any highly computable, $k-1$-regular connected graph with domatic number $k$ has a computable domatic $n$-partition.*

Actually, the graphs need NOT be connected.

Getting rid of the regularity requirement appears to be much harder.

Proving that we cannot make $k$ smaller also appears really difficult.

# In general. . .

**Proposition**

*For any $n$ and any $k \geq n^2 - n + 1$, any highly computable, $k-1$-regular connected graph with domatic number $k$ has a computable domatic $n$-partition.*

Actually, the graphs need NOT be connected.

Getting rid of the regularity requirement appears to be much harder.

Proving that we cannot make $k$ smaller also appears really difficult.

# In general...

**Proposition**

*For any $n$ and any $k \geq n^2 - n + 1$, any highly computable, $k-1$-regular connected graph with domatic number $k$ has a computable domatic $n$-partition.*

Actually, the graphs need NOT be connected.

Getting rid of the regularity requirement appears to be much harder.

Proving that we cannot make $k$ smaller also appears really difficult.

# Between computable and highly computable

In a computable graph, the neighborhood relation is computable from $K$.

In a highly computable graph, the neighborhood relation is computable.

What happens between these?

**Definition (Gasarch, Lee)**

A graph is $A$-computable provided the neighborhood relation is computable from $A$.

**Proposition**

*Let $A$ be any non-computable c.e. set. For all $n \geq 2$ there exists an $A$-computable graph $G$ with $d(G) = n$ but $d^c(G) = 2$.*

# Between computable and highly computable

In a computable graph, the neighborhood relation is computable from $K$.

In a highly computable graph, the neighborhood relation is computable.

What happens between these?

**Definition (Gasarch, Lee)**

A graph is $A$-computable provided the neighborhood relation is computable from $A$.

**Proposition**

*Let $A$ be any non-computable c.e. set. For all $n \geq 2$ there exists an $A$-computable graph $G$ with $d(G) = n$ but $d^c(G) = 2$.*

# Between computable and highly computable

In a computable graph, the neighborhood relation is computable from $K$.

In a highly computable graph, the neighborhood relation is computable.

What happens between these?

**Definition (Gasarch, Lee)**

A graph is $A$-computable provided the neighborhood relation is computable from $A$.

**Proposition**

*Let $A$ be any non-computable c.e. set. For all $n \geq 2$ there exists an $A$-computable graph $G$ with $d(G) = n$ but $d^c(G) = 2$.*

# The End

Thanks for listening