Minimum Sum Colorings of Chordal Graphs

Ian DeHaan and Zachary Friggstad

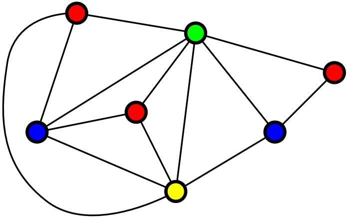
UNIVERSITY OF WATERLOO

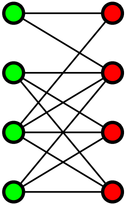
UNIVERSITY OF ALBERTA

WADS 2023

Graph Coloring

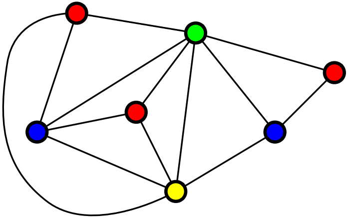
Adjacent vertices must receive different colors.

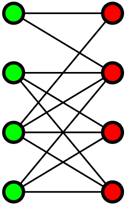




Graph Coloring

Adjacent vertices must receive different colors.

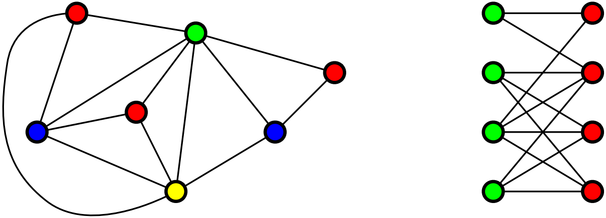




Chromatic Number: $\chi(G)$ = fewest colors required to color G.

Graph Coloring

Adjacent vertices must receive different colors.



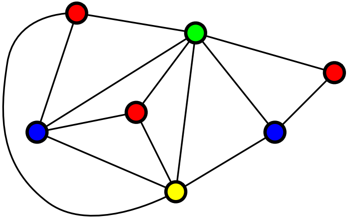
Chromatic Number: $\chi(G)$ = fewest colors required to color G.

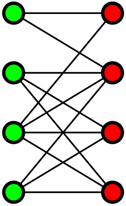
Scheduling Models

- Vertices: jobs to be processed
- Edges: resource conflicts
- Objective: minimize the final completion time

Graph Coloring

Adjacent vertices must receive different colors.





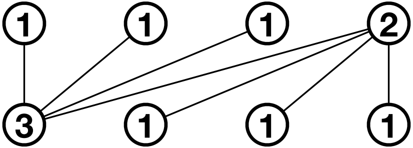
Chromatic Number: $\chi(G)$ = fewest colors required to color G.

Scheduling Models

- Vertices: jobs to be processed
- Edges: resource conflicts
- Objective: minimize the ~~final~~ **average** completion time

Minimum-Sum Coloring Problem

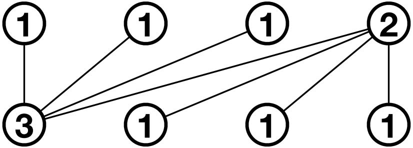
Find a coloring $\phi : V \to \{1, 2, 3, \ldots\}$ which minimizes $\sum_{v \in V} \phi(v)$.



Using 2 colors gives a minimum sum of **12**,

Minimum-Sum Coloring Problem

Find a coloring $\phi : V \to \{1, 2, 3, \ldots\}$ which minimizes $\sum_{v \in V} \phi(v)$.

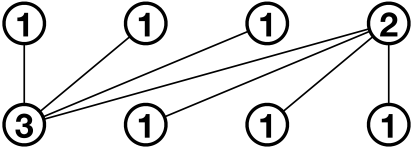


Using 2 colors gives a minimum sum of **12**,

but allowing 3 colors gives a minimum sum of **11**.

Minimum-Sum Coloring Problem

Find a coloring $\phi : V \to \{1, 2, 3, \ldots\}$ which minimizes $\sum_{v \in V} \phi(v)$.

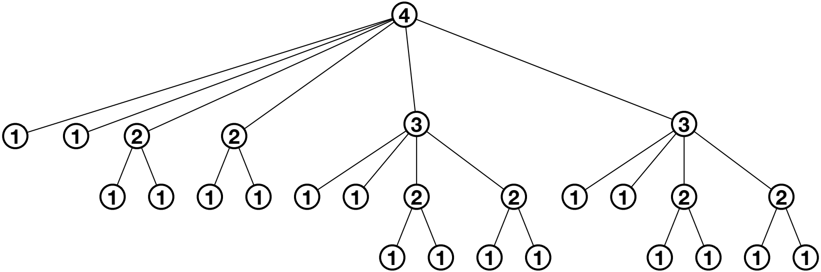


Using 2 colors gives a minimum sum of **12**,

but allowing 3 colors gives a minimum sum of **11**.

There is no $n^{0.999}$-approximation for arbitrary graphs.
[Bar-Noy et al, 1998]

Min-Sum Coloring on Trees



- The optimum coloring uses $O(\log n)$ colors.
  [Kubicka and Schwenk, 1989]
- An optimal solution can be computed in $O(n \cdot \log^2 n)$ time using dynamic programming.

Min-Sum Coloring on Bipartite Graphs

It is APX-HARD to compute an optimum min-sum coloring in bipartite graphs [Bar-Noy and Kortsarz, 1998],

Min-Sum Coloring on Bipartite Graphs

It is APX-HARD to compute an optimum min-sum coloring in bipartite graphs [Bar-Noy and Kortsarz, 1998],

but there is a $\frac{27}{26}$-approximation. [Malafiejski et al., 2004]

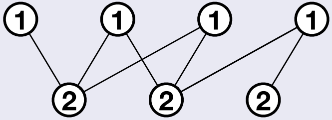
Min-Sum Coloring on Bipartite Graphs

It is APX-HARD to compute an optimum min-sum coloring in bipartite graphs [Bar-Noy and Kortsarz, 1998],

but there is a $\frac{27}{26}$-approximation. [Malafiejski et al., 2004]

Example:

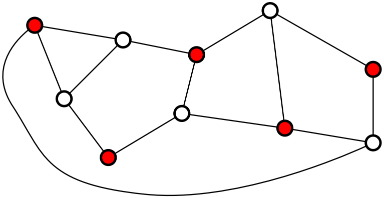
The natural 2-coloring is a 1.5-approximation.

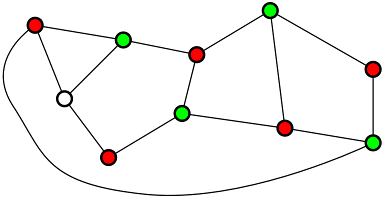


Analysis: The sum of colors is at most $n + \frac{1}{2} \cdot n = 1.5 \cdot n \leq 1.5 \cdot OPT$.

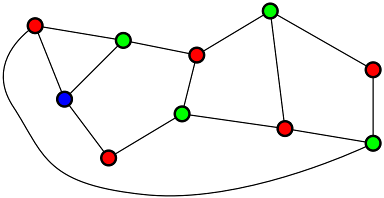
A Greedy Algorithm

Greedily coloring maximum independent sets yields a 4-approximation. [Bar-Noy et al, 1998]





Red = 1
Green = 2
Blue = 3



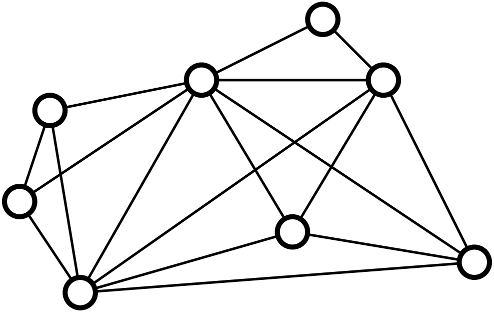
Some Known Approximation Bounds for Min-Sum coloring

Graph Class	Upper Bound	Lower Bound
Perfect	3.591	APX-HARD
Chordal	**1.796** $+ \epsilon$	APX-HARD
Interval	1.796	APX-HARD
Bipartite	27/26	APX-HARD
Planar	PTAS	NP-HARD
Line graphs	1.8298	APX-HARD

Chordal Graphs

Definition (Chordal Graphs)

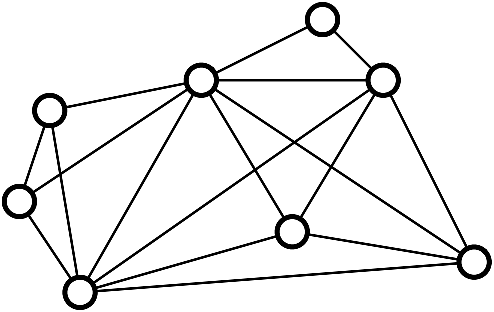
No induced cycles of length ≥ 4, a.k.a. triangulated graphs.



Chordal Graphs

Definition (Chordal Graphs)

No induced cycles of length ≥ 4, a.k.a. triangulated graphs.



95% of "interference" graphs in the Java 1.5 library are chordal.
[Pereira and Palsberg, 2005]

Interval Graph Algorithm

Suppose G allows us to compute a maximum-size k-colorable subgraph ($\textsc{MkCS}$) in polynomial time for any $k \geq 1$.

Then, a 1.796-approximation is possible. [Halldórsson et al., 2008]

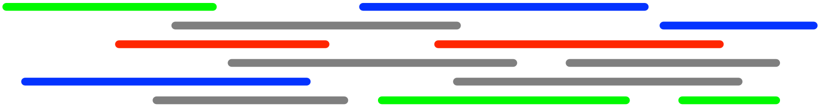


Figure: Interval graphs can be used to represent concurrent tasks.

Interval Graph Algorithm

Set $c := 3.591$ and pick some $\delta \in [0,1)$ uniformly randomly.

for $k = c^{\delta}, c^{\delta+1}, c^{\delta+2}, c^{\delta+3}, \ldots$

- Find a maximum k-colorable subgraph of G.
- Use the next k unused integers to color it.
- Remove these nodes from G.

Interval Graph Algorithm

Set $c := 3.591$ and pick some $\delta \in [0, 1)$ uniformly randomly.

for $k = c^{\delta}, c^{\delta+1}, c^{\delta+2}, c^{\delta+3}, \ldots$

- Find a maximum k-colorable subgraph of G.
- Use the next k unused integers to color it.
- Remove these nodes from G.

This yields an approximation with guarantee

$$\frac{c+1}{2 \cdot \ln c} \approx 1.796$$

MkCS in Chordal Graphs

Unfortunately, MkCS in chordal graphs is **NP**-hard.
[Yannakakis and Gavril, 1987]

MkCS in Chordal Graphs

Unfortunately, MkCS in chordal graphs is **NP**-hard. [Yannakakis and Gavril, 1987]

However, it can be approximated!

Theorem (D., Friggstad, 2023)

For any constant $\epsilon > 0$, there is a polynomial-time $(1 - \epsilon)$-approximation for MkCS in chordal graphs.

MkCS in Chordal Graphs

Unfortunately, MkCS in chordal graphs is **NP**-hard.
[Yannakakis and Gavril, 1987]

However, it can be approximated!

Theorem (D., Friggstad, 2023)

For any constant $\epsilon > 0$, there is a polynomial-time $(1 - \epsilon)$-approximation for MkCS in chordal graphs.

To get an improved approximation for min-sum coloring, we require an approach which can be used with MkCS approximations.

$\textsc{MkCS}$ in Chordal Graphs

Theorem (D., Friggstad, 2023)

Let $\mathcal{G}$ be a graph class that is closed under taking induced subgraphs. If there is a PTAS for weighted $\textsc{MkCS}$ in $\mathcal{G}$, then for any $\epsilon > 0$, there is a polynomial-time $(1.796 + \epsilon)$-approximation for minimum-sum coloring in $\mathcal{G}$.

MKCS in Chordal Graphs

Theorem (D., Friggstad, 2023)

Let $\mathcal{G}$ be a graph class that is closed under taking induced subgraphs. If there is a PTAS for weighted MKCS in $\mathcal{G}$, then for any $\epsilon > 0$, there is a polynomial-time $(1.796 + \epsilon)$-approximation for minimum-sum coloring in $\mathcal{G}$.

General Statement: Given a ρ-approximation for MkCS, our min-sum coloring approximation guarantee is:

$$\inf_{1 < c < \frac{1}{1-\rho}} \quad \frac{\rho \cdot (c+1)}{2 \cdot (1 - (1-\rho) \cdot c) \cdot \ln c}$$

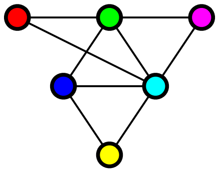
Previous Work

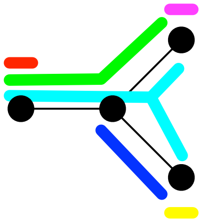
Weighted MkCS in chordal graphs can be solved in $n^{O(k)}$ time.
[Yannakakis and Gavril, 1987]

Previous Work

Weighted $\textsc{MkCS}$ in chordal graphs can be solved in $n^{O(k)}$ time.
[Yannakakis and Gavril, 1987]

Any chordal graph is the intersection graph of subtrees of a tree $\mathcal{T}$.

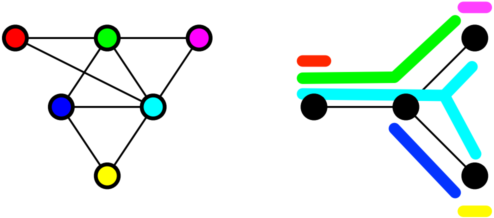




Previous Work

Weighted MKCS in chordal graphs can be solved in $n^{O(k)}$ time.
[Yannakakis and Gavril, 1987]

Any chordal graph is the intersection graph of subtrees of a tree $\mathcal{T}$.



Dynamic Program:

For $v \in \mathcal{T}$ and any $\leq k$ subtrees S spanning v:

$$f(v, S) = \text{best solution in the subtree under } v \text{ that uses } S \text{ at } v$$

Approximations for Large k

Theorem (D., Friggstad, 2023)

There is a $\mathrm{poly}(n)$*-time* $(1 - 2/k^{1/3})$*-approximation for weighted* MkCS *in chordal graphs.*

Approximations for Large k

Theorem (D., Friggstad, 2023)

There is a $\mathrm{poly}(n)$*-time* $(1 - 2/k^{1/3})$*-approximation for weighted* MkCS *in chordal graphs.*

To get the $(1 - \epsilon)$-approximation for any constant ϵ:

- If $k \leq 8/\epsilon^3$, use the exact algorithm (running time $n^{O(1/\epsilon^3)}$).
- Otherwise, use our new $(1 - 2/k^{1/3}) \geq (1 - \epsilon)$ approximation.

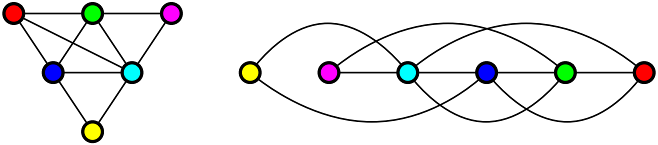
Perfect Elimination Orderings

An ordering $v_1, ..., v_n$ of the nodes such that for each v_i, its right neighbors $N^r(v_i)$ form a clique.

Perfect Elimination Orderings

An ordering $v_1, \ldots, v_n$ of the nodes such that for each v_i, its right neighbors $N^r(v_i)$ form a clique.

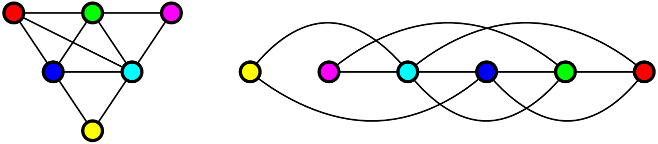
A graph is chordal iff it has a **perfect elimination ordering**.



Perfect Elimination Orderings

An ordering $v_1, ..., v_n$ of the nodes such that for each v_i, its right neighbors $N^r(v_i)$ form a clique.

A graph is chordal iff it has a **perfect elimination ordering**.

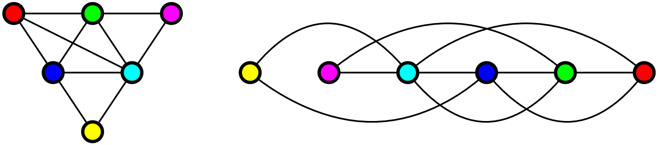


This ordering can be computed in linear time.

Perfect Elimination Orderings

An ordering $v_1, \ldots, v_n$ of the nodes such that for each v_i, its right neighbors $N^r(v_i)$ form a clique.

A graph is chordal iff it has a **perfect elimination ordering**.



This ordering can be computed in linear time.

Remark

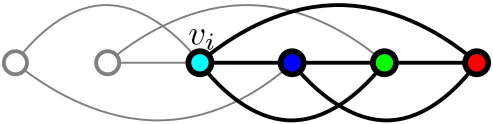
$S \subseteq V$ induces a k-colorable subgraph iff for each $v_i \in S$,

$$|N^r(v_i) \cap S| \leq k - 1.$$

Linear Programming Formulation

Let x_i indicate if we select v_i in our k-colorable subgraph.

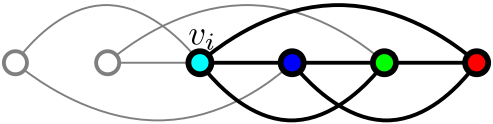
$$\begin{array}{rrcll} \textbf{maximize}: & \sum_i w_i \cdot x_i & & & \\ \textbf{subject to}: & x_i + \sum_{v_j \in N^r(v_i)} x_j & \leq & k & \forall\ i \\ & x_i & \in & [0,1] & \forall\ i \end{array}$$



Linear Programming Formulation

Let x_i indicate if we select v_i in our k-colorable subgraph.

$$
\begin{array}{rrcll}
\textbf{maximize}: & \sum_i w_i \cdot x_i & & & \\
\textbf{subject to}: & x_i + \sum_{v_j \in N^r(v_i)} x_j & \leq & k & \forall\ i \\
 & x_i & \in & [0,1] & \forall\ i
\end{array}
$$



Rounding Algorithm

- $S \leftarrow \varnothing$
- **for** i from n down to 1:
  - Flip a coin c_i with bias $(1 - 1/k^{1/3}) \cdot x_i$ towards heads.
  - If $S \cup \{v_i\}$ is feasible and c_i = heads, add v_i to S.

Analysis of the Rounding Algorithm

For each v_i, the **expected** number of `heads` flipped from $N^r(v_i)$ is at most $k \cdot (1 - k^{-1/3}) = k - k^{2/3}$.

Analysis of the Rounding Algorithm

For each v_i, the **expected** number of `heads` flipped from $N^r(v_i)$ is at most $k \cdot (1 - k^{-1/3}) = k - k^{2/3}$.

The probability that at least k coins from $N^r(v_i)$ are heads is at most $k^{-1/3}$ (Chebyshev's inequality).

Analysis of the Rounding Algorithm

For each v_i, the **expected** number of heads flipped from $N^r(v_i)$ is at most $k \cdot (1 - k^{-1/3}) = k - k^{2/3}$.

The probability that at least k coins from $N^r(v_i)$ are heads is at most $k^{-1/3}$ (Chebyshev's inequality).

Thus, v_i will be added with probability at least:

$$\mathbf{Pr}[|\#\texttt{heads} \text{ from } N^r(v_i)| < k \wedge c_i = \texttt{heads}] \geq (1-k^{-1/3}) \cdot (1-k^{-1/3}) \cdot x_i.$$

Analysis of the Rounding Algorithm

For each v_i, the **expected** number of `heads` flipped from $N^r(v_i)$ is at most $k \cdot (1 - k^{-1/3}) = k - k^{2/3}$.

The probability that at least k coins from $N^r(v_i)$ are heads is at most $k^{-1/3}$ (Chebyshev's inequality).

Thus, v_i will be added with probability at least:

$$\mathbf{Pr}\left[|\#\texttt{heads from } N^r(v_i)| < k \wedge c_i = \texttt{heads}\right] \geq (1-k^{-1/3})\cdot(1-k^{-1/3})\cdot x_i.$$

$\therefore$ The expected size of S is at least $(1 - 2/k^{1/3}) \cdot OPT_{LP}$.

Analysis of the Rounding Algorithm

This analysis approach was presented because there are generic ways to efficiently de-randomize such algorithms if the analysis only uses second moments.

Analysis of the Rounding Algorithm

This analysis approach was presented because there are generic ways to efficiently de-randomize such algorithms if the analysis only uses second moments.

With more hands-on work, we can de-randomize tighter analysis (using Chernoff bounds) to get a deterministic algorithm with guarantee

$$1 - \frac{O(1)}{k \log k}.$$

k-Colorable Subgraphs to Min-Sum Coloring

Theorem (D., Friggstad 2023)

Let $\mathcal{G}$ be a graph class that is closed under taking induced subgraphs.If there is a PTAS for weighted MkCS *in $\mathcal{G}$, then for any $\epsilon > 0$, there is a polynomial-time $(1.796 + \epsilon)$-approximation for minimum-sum coloring in $\mathcal{G}$.*

Linear Program

- $x_{v,k}$ - indicates that v has color k.
- $z_{C,k}$ - indicates that C is the set of nodes that are $\leq k$-colored.

$$
\begin{aligned}
\textbf{minimize:} \quad & \sum_{v \in V} \sum_{k=1}^{n} w_v \cdot k \cdot x_{v,k} && && (1) \\
\textbf{subject to:} \quad & \sum_{k=1}^{n} x_{v,k} = 1 && \forall\ v \in V && (2) \\
& \sum_{C \in \mathcal{C}_k} z_{C,k} \leq 1 && \forall\ 1 \leq k \leq n && (3) \\
& \sum_{C \in \mathcal{C}_k : v \in C} z_{C,k} \geq \sum_{k' \leq k} x_{v,k'} && \forall\ v \in V, 1 \leq k \leq n && (4) \\
& x, z \geq 0
\end{aligned}
$$

Last constraint: (partial) agreement between z and x on the statement v is colored by color at most k.

Taking the Dual

maximize: $$\sum_{v \in V} \alpha_v - \sum_{k=1}^{n} \beta_k$$

subject to:

$$\alpha_v \leq w_v \cdot k + \sum_{\hat{k}=k}^{n} \theta_{v,\hat{k}} \quad \forall\ v \in V, 1 \leq k \leq n \tag{5}$$

$$\sum_{v \in C} \theta_{v,k} \leq \beta_k \quad \forall\ 1 \leq k \leq n, C \in \mathcal{C}_k \tag{6}$$

$$\beta, \theta \geq 0 \tag{7}$$

Solving and Rounding

Via the ellipsoid method for solving LPs, this yields a solution (x, z) with value $\leq OPT$ for the following slightly modified LP.

$$\begin{array}{rcll} \sum_t x_{v,t} & = & 1 & \forall\ v \\ \sum_S z_{S,t} & = & 1/\rho & \forall\ t \\ \sum_{S \ni v} z_{S,t} & \geq & \sum_{t' \leq t} x_{t',v} & \forall\ v, t \\ x, z & \geq & 0 & \end{array}$$

Rounding Algorithm

$c \leftarrow 3.591$

$\delta \sim [0, 1)$ uniformly at random

For $k := c^{\delta}, c^{\delta+1}, c^{\delta+2}, \ldots$

- Sample a $\lfloor k \rfloor$-colorable subset S from the distribution $\rho \cdot z_S$.
- Randomly permute the coloring.
- Concatenate this coloring to the coloring of G so far.

Next Steps: Min-Sum Coloring on Perfect Graphs

Min-sum coloring in perfect graphs has a 3.592-approximation. [Gandhi et al., 2008]

Next Steps: Min-Sum Coloring on Perfect Graphs

Min-sum coloring in perfect graphs has a 3.592-approximation. [Gandhi et al., 2008]

Question:

Can we do better using our framework?

Next Steps: Min-Sum Coloring on Perfect Graphs

Min-sum coloring in perfect graphs has a 3.592-approximation. [Gandhi et al., 2008]

Question:

Can we do better using our framework?

MkCS is APX-HARD in perfect graphs, even for $k = 2$. [Addario-Berry et al., 2010]

Next Steps: Min-Sum Coloring on Perfect Graphs

Min-sum coloring in perfect graphs has a 3.592-approximation.
[Gandhi et al., 2008]

Question:

Can we do better using our framework?

MkCS is APX-HARD in perfect graphs, even for $k = 2$.
[Addario-Berry et al., 2010]

We need a 0.704-approx to do better using our framework; the best known is a 0.632-approx.

Next Steps: Min-Sum Coloring on Perfect Graphs

Min-sum coloring in perfect graphs has a 3.592-approximation. [Gandhi et al., 2008]

Question:

Can we do better using our framework?

MkCS is APX-HARD in perfect graphs, even for $k = 2$. [Addario-Berry et al., 2010]

We need a 0.704-approx to do better using our framework; the best known is a 0.632-approx.

Next Steps: Latency Constant

The "latency constant" $c \approx 3.591$ shows up in the current best approximation ratio for interval, chordal, and perfect graphs.

Next Steps: Latency Constant

The “latency constant” $c \approx 3.591$ shows up in the current best approximation ratio for interval, chordal, and perfect graphs.

It also shows up in some best known approximations for minimum latency problems.

Next Steps: Latency Constant

The “latency constant” $c \approx 3.591$ shows up in the current best approximation ratio for interval, chordal, and perfect graphs.

It also shows up in some best known approximations for minimum latency problems.

Question:

Is c a fundamental lower bound, or can these approximations be improved?

Thank you!