

184.754 Seminar on Algorithms

Paper: Coloring the Vertices of 9-pt and 27-pt
Stencils with Intervals

Supervisor: Prof. Dr. Jesper Larsson Träff

Camilo Tello Fachin
December 7, 2023

Introduction

Interval Vertex Coloring (IVC) Definition

Special Case Analysis

Definitions for Analysis

Special Cases

Heuristics

NP - Completeness

Algorithms

Application and Experiments

References

Connection between VCP and Parallel Algorithms

Connection between VCP and Parallel Algorithms

- Processors could be colors while the vertices could be tasks.

Connection between VCP and Parallel Algorithms

- Processors could be colors while the vertices could be tasks.
- What if tasks are not equally computationally expensive?

Connection between VCP and Parallel Algorithms

- Processors could be colors while the vertices could be tasks.
- What if tasks are not equally computationally expensive?
- ↫ add weights \propto cost and solve IVCP!

Connection between VCP and Parallel Algorithms

- Processors could be colors while the vertices could be tasks.
- What if tasks are not equally computationally expensive?
- ↪ add weights \propto cost and solve IVCP!
- If weights are good estimate, better performance.

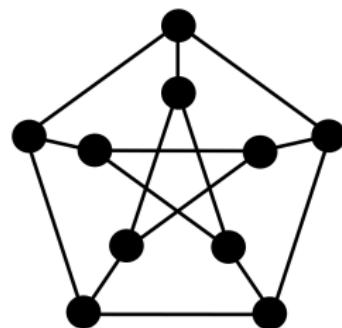
Connection between VCP and Parallel Algorithms

- Processors could be colors while the vertices could be tasks.
- What if tasks are not equally computationally expensive?
- ↪ add weights \propto cost and solve IVCP!
- If weights are good estimate, better performance.
- optimal (minimal) colorings in such an application, can lead to runtime improvement \propto degree of parallelism.

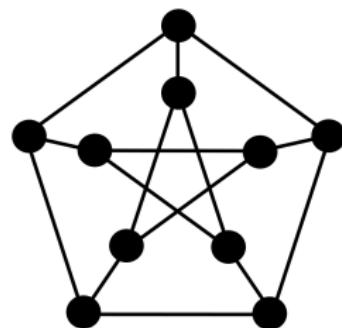
Connection between VCP and Parallel Algorithms

- Processors could be colors while the vertices could be tasks.
- What if tasks are not equally computationally expensive?
- ↪ add weights \propto cost and solve IVCP!
- If weights are good estimate, better performance.
- optimal (minimal) colorings in such an application, can lead to runtime improvement \propto degree of parallelism.
- Unfortunately problem is NP-Complete, heuristics are required for large instances to get useful solutions.

- Given $G(V, E)$

 $G(V, E)$ 

- Given $G(V,E)$
- Find a vertex coloring s.t. colors on adjacent vertices differ

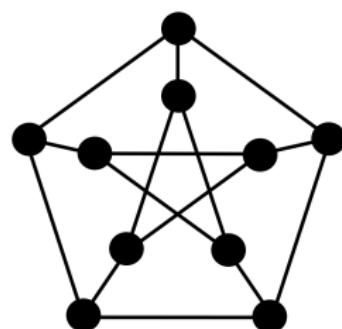
 $G(V,E)$ 

- Given $G(V, E)$
- Find a vertex coloring s.t. colors on adjacent vertices differ

Formal Definition of VCP given
 $G(V, E)$

find $f(v)$:
 $\forall v \in V : \forall w \text{ in } \Gamma(v) : f(v) \neq f(w).$

$G(V, E)$

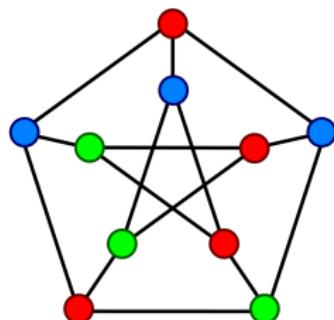


- Given $G(V, E)$
- Find a vertex coloring s.t. colors on adjacent vertices differ

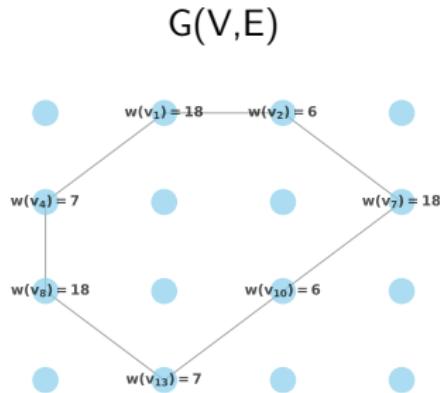
Formal Definition of VCP given
 $G(V, E)$

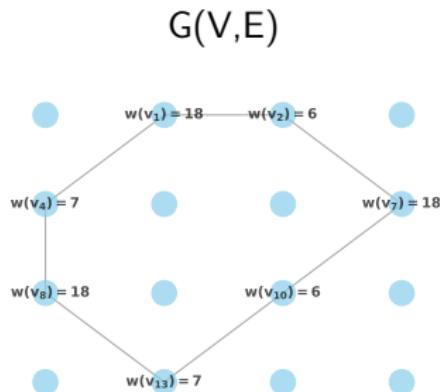
find $f(v)$:
 $\forall v \in V : \forall w \text{ in } \Gamma(v) : f(v) \neq f(w).$

$G(V, E)$
Colored

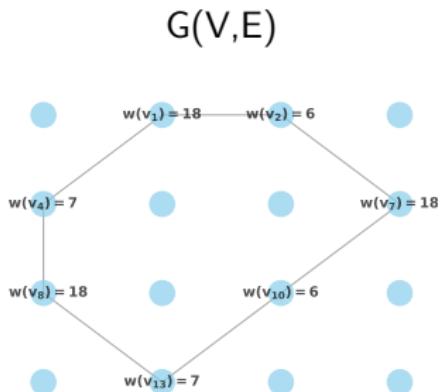


- Let $G(V,E)$ be an undirected graph and $w : V \mapsto \mathbb{Z}^+$ the weights.

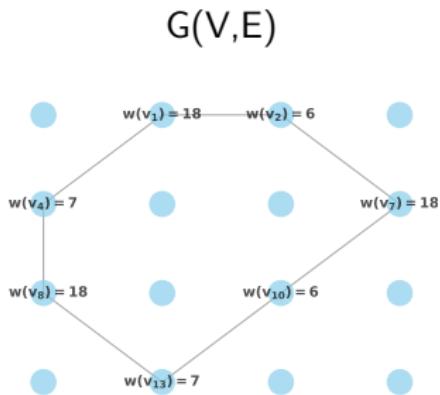




- Let $G(V, E)$ be an undirected graph and $w : V \mapsto \mathbb{Z}^+$ the weights.
- A interval coloring of the vertices of G is a function $\text{start} : V \mapsto \mathbb{Z}^+$



- Let $G(V, E)$ be an undirected graph and $w : V \mapsto \mathbb{Z}^+$ the weights.
- A interval coloring of the vertices of G is a function $\text{start} : V \mapsto \mathbb{Z}^+$
- Vertex v is colored with open interval: $[\text{start}(v), \text{start}(v) + w(v))$



- Let $G(V, E)$ be an undirected graph and $w : V \mapsto \mathbb{Z}^+$ the weights.
- A interval coloring of the vertices of G is a function $\text{start} : V \mapsto \mathbb{Z}^+$
- Vertex v is colored with open interval: $[\text{start}(v), \text{start}(v) + w(v))$
- Neighboring Vertices must have disjoint color intervals: $\forall (a, b) \in E : [\text{start}(a), \text{start}(a) + w(a)) \cap [\text{start}(b), \text{start}(b) + w(b)) = \emptyset$.

- A interval coloring of the vertices of G is a function $\text{start} : V \mapsto \mathbb{Z}^+$

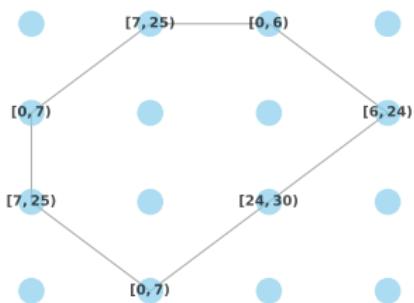
- A interval coloring of the vertices of G is a function $\text{start} : V \mapsto \mathbb{Z}^+$

- A particular coloring "start" of the vertices of G is said to use:

$$\text{maxcolor} = \max_{v \in V} \text{start}(v) + w(v) \text{ colors.}$$

- A interval coloring of the vertices of G is a function $\text{start} : V \mapsto \mathbb{Z}^+$

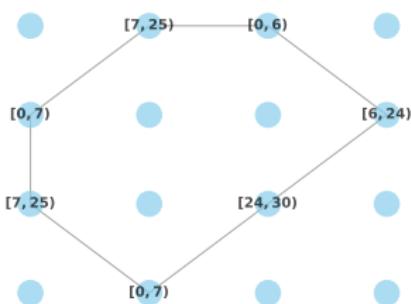
Interval Colored $G(V, E)$
with $\text{maxcolor} = 30$



- A particular coloring "start" of the vertices of G is said to use:

$$\text{maxcolor} = \max_{v \in V} \text{start}(v) + w(v) \text{ colors.}$$

Interval Colored $G(V, E)$
with $\text{maxcolor} = 30$



- A interval coloring of the vertices of G is a function $\text{start} : V \mapsto \mathbb{Z}^+$

- A particular coloring "start" of the vertices of G is said to use:

$$\text{maxcolor} = \max_{v \in V} \text{start}(v) + w(v) \text{ colors.}$$

Optimization Problem Instance:

Find a coloring $\text{start} : V \mapsto \mathbb{Z}^+$ that minimizes maxcolor .

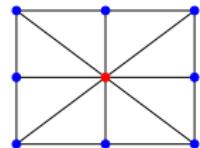
- A maxcolor that is indeed minimal, is denoted with maxcolor^* .

Definition: 2D Stencil Interval Vertex Coloring (2DS-IVC)

A problem where G is a 9-pt 2D stencil, composed of $X \times Y$ vertices on a 2D grid such that two vertices (i, j) and (i', j') are connected iff $|i - i'| \leq 1$ and $|j - j'| \leq 1$.

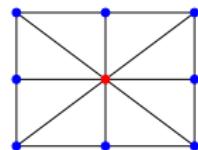
Definition: 2D Stencil Interval Vertex Coloring (2DS-IVC)

A problem where G is a 9-pt 2D stencil, composed of $X \times Y$ vertices on a 2D grid such that two vertices (i, j) and (i', j') are connected iff $|i - i'| \leq 1$ and $|j - j'| \leq 1$.



Definition: 2D Stencil Interval Vertex Coloring (2DS-IVC)

A problem where G is a 9-pt 2D stencil, composed of $X \times Y$ vertices on a 2D grid such that two vertices (i, j) and (i', j') are connected iff $|i - i'| \leq 1$ and $|j - j'| \leq 1$.

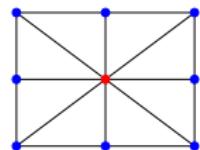


Definition: 3D Stencil Interval Vertex Coloring (3DS-IVC)

A problem where G is a 27-point 3D stencil, composed of $X \times Y \times Z$ vertices on a 3D grid such that two vertices (i, j, k) and (i', j', k') are connected iff $|i - i'| \leq 1$, $|j - j'| \leq 1$, and $|k - k'| \leq 1$.

Definition: 2D Stencil Interval Vertex Coloring (2DS-IVC)

A problem where G is a 9-pt 2D stencil, composed of $X \times Y$ vertices on a 2D grid such that two vertices (i, j) and (i', j') are connected iff $|i - i'| \leq 1$ and $|j - j'| \leq 1$.



Definition: 3D Stencil Interval Vertex Coloring (3DS-IVC)

A problem where G is a 27-point 3D stencil, composed of $X \times Y \times Z$ vertices on a 3D grid such that two vertices (i, j, k) and (i', j', k') are connected iff $|i - i'| \leq 1$, $|j - j'| \leq 1$, and $|k - k'| \leq 1$.

Theorem

Let $G(V, E)$ be a clique K_n with n vertices, then

$$\text{maxcolor} = \sum_{v \in V} w(v)$$

is indeed an optimal coloring maxcolor^* .

Theorem

Let $G(V, E)$ be a clique K_n with n vertices, then

$$\text{maxcolor} = \sum_{v \in V} w(v)$$

is indeed an optimal coloring maxcolor^* .

- There exists such a coloring by greedily allocating the color interval with the lowest available $\text{start}(v)$ with $\mathcal{O}(n)$ □.

Theorem

Let $G(V, E)$ be a clique K_n with n vertices, then

$$\text{maxcolor} = \sum_{v \in V} w(v)$$

is indeed an optimal coloring maxcolor^* .

- There exists such a coloring by greedily allocating the color interval with the lowest available start(v) with $\mathcal{O}(n)$ \square .
- Particularly important result since the 2D-IVC's and 3D-IVC's are composed of K_4 's and K_8 's respectively.

Theorem

Let $G(V, E)$ be a clique K_n with n vertices, then

$$\text{maxcolor} = \sum_{v \in V} w(v)$$

is indeed an optimal coloring maxcolor^* .

- There exists such a coloring by greedily allocating the color interval with the lowest available $\text{start}(v)$ with $\mathcal{O}(n)$ \square .
- Particularly important result since the 2D-IVC's and 3D-IVC's are composed of K_4 's and K_8 's respectively.
- Therefore any identified K_4 in a 2D-IVC Problem and any K_8 in a 3D-IVC Problem are (lose) lower bounds on maxcolor.

Theorem

A graph $G(V,E)$ is Bipartite iff it contains no odd cycles. In other words: its vertices can be partitioned into two sets A and B , such that all edges have one endpoint in A and the other in B .

Theorem

A graph $G(V,E)$ is Bipartite iff it contains no odd cycles. In other words: its vertices can be partitioned into two sets A and B , such that all edges have one endpoint in A and the other in B .

- All edges provide trivial lower bound:
 $\text{maxcolor}^* \geq w(i) + w(j), \forall (i,j) \in E.$

Theorem

A graph $G(V,E)$ is Bipartite iff it contains no odd cycles. In other words: its vertices can be partitioned into two sets A and B , such that all edges have one endpoint in A and the other in B .

- All edges provide trivial lower bound:
 $\text{maxcolor}^* \geq w(i) + w(j), \forall (i,j) \in E.$
- There exists a coloring
 $\text{maxcolor}^* = \max_{(i,j) \in E} w(i) + w(j)$ in $\mathcal{O}(|E|)$. (Proof Omitted)

Theorem

A graph $G(V,E)$ is Bipartite iff it contains no odd cycles. In other words: its vertices can be partitioned into two sets A and B , such that all edges have one endpoint in A and the other in B .

- All edges provide trivial lower bound:
 $\text{maxcolor}^* \geq w(i) + w(j), \forall (i,j) \in E.$
- There exists a coloring
 $\text{maxcolor}^* = \max_{(i,j) \in E} w(i) + w(j)$ in $\mathcal{O}(|E|)$. (Proof Omitted)
- Another important result, since the 9-pt stencil contains a Bipartite 5-pt stencil and the 27-pt stencil contains at least a Bipartite 7-pt stencil, and all paths are Bipartite.

Theorem

A graph $G(V,E)$ is Bipartite iff it contains no odd cycles. In other words: its vertices can be partitioned into two sets A and B , such that all edges have one endpoint in A and the other in B .

- All edges provide trivial lower bound:
 $\text{maxcolor}^* \geq w(i) + w(j), \forall (i,j) \in E.$
- There exists a coloring
 $\text{maxcolor}^* = \max_{(i,j) \in E} w(i) + w(j)$ in $\mathcal{O}(|E|)$. (Proof Omitted)
- Another important result, since the 9-pt stencil contains a Bipartite 5-pt stencil and the 27-pt stencil contains at least a Bipartite 7-pt stencil, and all paths are Bipartite.
- Authors use this to construct approximation algorithms.

- Graphs that are not Bipartite, contain at least one cycle of odd length.

- Graphs that are not Bipartite, contain at least one cycle of odd length.
- Odd cycles have optimal interval colorings that are strictly greater than the largest weight of any clique in the cycle.

- Graphs that are not Bipartite, contain at least one cycle of odd length.
- Odd cycles have optimal interval colorings that are strictly greater than the largest weight of any clique in the cycle.
- The clique with the largest weight from the cycle shown earlier would be of interval size 25, while the optimal coloring of the entire graph (the odd cycle) has $\text{maxcolor}^* = 30$.

- Graphs that are not Bipartite, contain at least one cycle of odd length.
- Odd cycles have optimal interval colorings that are strictly greater than the largest weight of any clique in the cycle.
- The clique with the largest weight from the cycle shown earlier would be of interval size 25, while the optimal coloring of the entire graph (the odd cycle) has $\text{maxcolor}^* = 30$.
- Understanding of coloring of odd cycles yields new lower bounds.

Definition

Let maxpair be the maximal sum of any two consecutive vertices.¹ ²

$$\text{maxpair} = \max_i w(i, i + 1)$$

¹Here G is cycle, neighbors of vertex x are $x + 1$ and $x - 1$.

²The notation $w(x)$ is slightly abused s.t. $w(a, b, c) = w(a) + w(b) + w(c)$

Definition

Let maxpair be the maximal sum of any two consecutive vertices. ^{1 2}

$$\text{maxpair} = \max_i w(i, i + 1)$$

Definition

Let minchain3 be the minimum sum of any 3 consecutive vertices:

$$\text{minchain3} = \min_i w(i, i + 1, i + 2)$$

¹Here G is cycle, neighbors of vertex x are $x + 1$ and $x - 1$.

²The notation $w(x)$ is slightly abused s.t. $w(a, b, c) = w(a) + w(b) + w(c)$

Definition

Let maxpair be the maximal sum of any two consecutive vertices.¹ ²

$$\text{maxpair} = \max_i w(i, i + 1)$$

Definition

Let minchain3 be the minimum sum of any 3 consecutive vertices:

$$\text{minchain3} = \min_i w(i, i + 1, i + 2)$$

Theorem

Let G be an odd cycle, then it holds:

$$\text{maxcolor}^* = \max(\text{maxpair}, \text{minchain3})$$

¹Here G is cycle, neighbors of vertex x are $x + 1$ and $x - 1$.

²The notation $w(x)$ is slightly abused s.t. $w(a, b, c) = w(a) + w(b) + w(c)$

Idea of Proof

Idea of Proof

- If G is an odd cycle, there is an algorithm that yields $\max(\text{maxpair}, \text{minchain3})$ colors, which means $\text{maxcolor}^* \leq \max(\text{maxpair}, \text{minchain3})$.

Idea of Proof

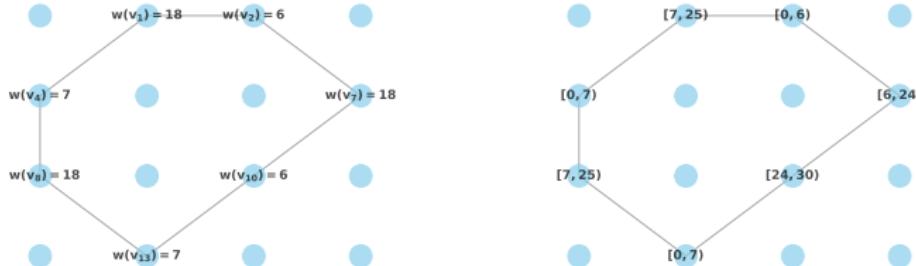
- If G is an odd cycle, there is an algorithm that yields $\max(\text{maxpair}, \text{minchain3})$ colors, which means $\text{maxcolor}^* \leq \max(\text{maxpair}, \text{minchain3})$.
- If G is an odd cycle, then $\text{maxcolor}^* \geq \max(\text{maxpair}, \text{minchain3})$.

Idea of Proof

- If G is an odd cycle, there is an algorithm that yields $\max(\text{maxpair}, \text{minchain3})$ colors, which means $\text{maxcolor}^* \leq \max(\text{maxpair}, \text{minchain3})$.
- If G is an odd cycle, then $\text{maxcolor}^* \geq \max(\text{maxpair}, \text{minchain3})$.
- Coinciding bounds conclude proof [1] \square .

Idea of Proof

- If G is an odd cycle, there is an algorithm that yields $\max(\text{maxpair}, \text{minchain3})$ colors, which means $\text{maxcolor}^* \leq \max(\text{maxpair}, \text{minchain3})$.
- If G is an odd cycle, then $\text{maxcolor}^* \geq \max(\text{maxpair}, \text{minchain3})$.
- Coinciding bounds conclude proof [1] \square .
- One can verify this for the previous example



Lemma

 $3DS\text{-}IVC \in NP$

Lemma

$3DS-IVC \in NP$

Proof.

Given a solution for 3DS-IVC, encoded with pairs of integers which are bounded between 0 and $\sum_{v \in V} w(v_i)$ \mapsto are solutions in polynomial space.

Lemma

$3DS-IVC \in NP$

Proof.

Given a solution for 3DS-IVC, encoded with pairs of integers which are bounded between 0 and $\sum_{v \in V} w(v_i) \mapsto$ are solutions in polynomial space.

Verify that no adjacent edges have overlapping intervals.

$\forall (u, v) \in E :$

$$[\text{start}(u), \text{start}(u) + w(u)] \cap [\text{start}(v), \text{start}(v) + w(v)] = \emptyset.$$

Lemma

$3DS\text{-}IVC \in NP$

Proof.

Given a solution for 3DS-IVC, encoded with pairs of integers which are bounded between 0 and $\sum_{v \in V} w(v_i)$ are solutions in polynomial space.

Verify that no adjacent edges have overlapping intervals.

$\forall (u, v) \in E :$

$$[\text{start}(u), \text{start}(u) + w(u)] \cap [\text{start}(v), \text{start}(v) + w(v)] = \emptyset.$$

Since $|E| \leq \frac{n(n-1)}{2}$ and checking if two intervals overlap is in $\mathcal{O}(1)$, the verification can be done in $\mathcal{O}(n^2)$ □.

Idea of Proof

- Construct an instance 3DS-IVC from an instance of NAE-3SAT in polynomial time.

Idea of Proof

- Construct an instance 3DS-IVC from an instance of NAE-3SAT in polynomial time.
- Verify that a positive instance of NAE-3SAT results in a positive instance of 3DS-IVC.

Idea of Proof

- Construct an instance 3DS-IVC from an instance of NAE-3SAT in polynomial time.
- Verify that a positive instance of NAE-3SAT results in a positive instance of 3DS-IVC.
- Verify that if the created instance of 3DS-IVC is positive, then the instance of NAE-3SAT is also positive.

Idea of Proof

- Construct an instance 3DS-IVC from an instance of NAE-3SAT in polynomial time.
- Verify that a positive instance of NAE-3SAT results in a positive instance of 3DS-IVC.
- Verify that if the created instance of 3DS-IVC is positive, then the instance of NAE-3SAT is also positive.
- Since the 3DS-IVC problem is in NP and is harder than NAE-3SAT, which is an NP-Complete problem, 3DS-IVC is NP-Complete

□.

Idea of Proof

- Construct an instance 3DS-IVC from an instance of NAE-3SAT in polynomial time.
- Verify that a positive instance of NAE-3SAT results in a positive instance of 3DS-IVC.
- Verify that if the created instance of 3DS-IVC is positive, then the instance of NAE-3SAT is also positive.
- Since the 3DS-IVC problem is in NP and is harder than NAE-3SAT, which is an NP-Complete problem, 3DS-IVC is NP-Complete

□.

The proof yields:

Theorem

Deciding whether a 27-pt stencil can be colored with less than k colors is NP-Complete. [1]

Recall: Theorem

For bipartite graphs there exists a coloring
 $\text{maxcolor}^* = \max_{(i,j) \in E} w(i) + w(j)$ attainable in $\mathcal{O}(|E|)$.

Recall: Theorem

For bipartite graphs there exists a coloring

$\text{maxcolor}^* = \max_{(i,j) \in E} w(i) + w(j)$ attainable in $\mathcal{O}(|E|)$.

Definition

Let $c(x, y)$ be the lower end of the color interval associated with vertex (x, y) in that coloring. And let $RC = \max c(x, y) + w(x, y)$ be the maximum color used by any of the rows.

Recall: Theorem

For bipartite graphs there exists a coloring

$\text{maxcolor}^* = \max_{(i,j) \in E} w(i) + w(j)$ attainable in $\mathcal{O}(|E|)$.

Definition

Let $c(x, y)$ be the lower end of the color interval associated with vertex (x, y) in that coloring. And let $RC = \max c(x, y) + w(x, y)$ be the maximum color used by any of the rows.

Corollary

$RC \leq \text{maxcolor}^*$ is a lower bound of the optimal number of colors of the instance since it is the optimal coloring of a subgraph of the original instance.

1 Algorithm: Bipartite Decomposition

Data: 2DS-IVC instance with Y rows

Result: Approximate coloring with at most $2 \cdot \text{maxcolor}^*$

```
2 for each row  $r$  in the 2DS-IVC instance do
3   if  $r$  is even then
4     L Color the chain of vertices in row  $r$  optimally using colors from  $[0, RC)$ ;
5   else
6     L Color the chain of vertices in row  $r$  optimally using colors from  $[RC, 2RC)$ ;
7 return Approximate coloring with at most  $2 \cdot \text{maxcolor}^*$ 
```

1 Algorithm: Bipartite Decomposition

Data: 2DS-IVC instance with Y rows
Result: Approximate coloring with at most $2 \cdot \text{maxcolor}^*$

```
2 for each row  $r$  in the 2DS-IVC instance do
3   if  $r$  is even then
4     L Color the chain of vertices in row  $r$  optimally using colors from  $[0, RC)$ ;
5   else
6     L Color the chain of vertices in row  $r$  optimally using colors from  $[RC, 2RC)$ ;
7 return Approximate coloring with at most  $2 \cdot \text{maxcolor}^*$ 
```

Theorem

Since the coloring obtained by the algorithm is at most $2 \cdot \text{maxcolor}^$, the Bipartite Decomposition is a 2-approximation algorithm for 2DS-IVC. [1]*

1 Algorithm: Bipartite Decomposition

Data: 2DS-IVC instance with Y rows
Result: Approximate coloring with at most $2 \cdot \text{maxcolor}^*$

```
2 for each row  $r$  in the 2DS-IVC instance do
3   if  $r$  is even then
4     L Color the chain of vertices in row  $r$  optimally using colors from  $[0, RC)$ ;
5   else
6     L Color the chain of vertices in row  $r$  optimally using colors from  $[RC, 2RC)$ ;
7 return Approximate coloring with at most  $2 \cdot \text{maxcolor}^*$ 
```

Theorem

Since the coloring obtained by the algorithm is at most $2 \cdot \text{maxcolor}^$, the Bipartite Decomposition is a 2-approximation algorithm for 2DS-IVC. [1]*

Theorem

the Bipartite Decomposition is a 4-approximation algorithm for 3DS-IVC. (same approach) [1]

Space-Time Kernel Density Estimation

$$f(x, t) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}, \frac{t - t_i}{h_t}\right)$$

- Parallelized summation leads to race condition.
- Implementation of space-time decomposition by Hohl et. al. [2].
- Parallelization by exploiting symmetry by Saule et. al. [3].

Space-Time Kernel Density Estimation

$$f(x, t) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}, \frac{t - t_i}{h_t}\right)$$

- Parallelized summation leads to race condition.
- Implementation of space-time decomposition by Hohl et. al. [2].
- Parallelization by exploiting symmetry by Saule et. al. [3].

Authors Approach

Space-Time Kernel Density Estimation

$$f(x, t) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}, \frac{t - t_i}{h_t}\right)$$

- Parallelized summation leads to race condition.
- Implementation of space-time decomposition by Hohl et. al. [2].
- Parallelization by exploiting symmetry by Saule et. al. [3].

Authors Approach

- Arbitrary decomposition of space-time domain.

Space-Time Kernel Density Estimation

$$f(x, t) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}, \frac{t - t_i}{h_t}\right)$$

- Parallelized summation leads to race condition.
- Implementation of space-time decomposition by Hohl et. al. [2].
- Parallelization by exploiting symmetry by Saule et. al. [3].

Authors Approach

- Arbitrary decomposition of space-time domain.
- 2 spatial coordinates, 1 time coordinate \mapsto 3DS-IVC Problem

Space-Time Kernel Density Estimation

$$f(x, t) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}, \frac{t - t_i}{h_t}\right)$$

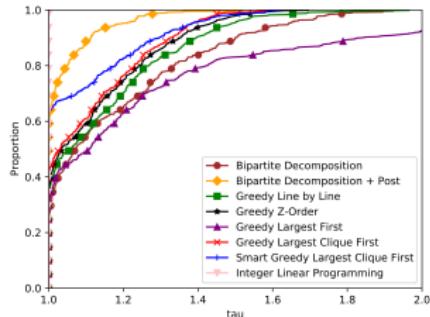
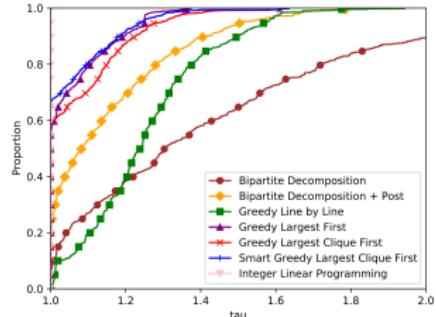
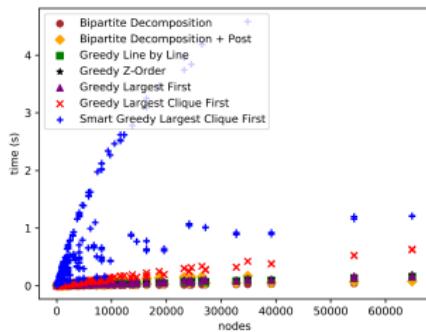
- Parallelized summation leads to race condition.
- Implementation of space-time decomposition by Hohl et. al. [2].
- Parallelization by exploiting symmetry by Saule et. al. [3].

Authors Approach

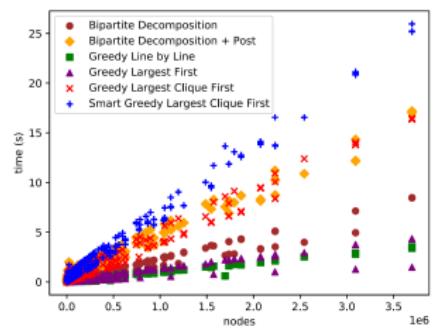
- Arbitrary decomposition of space-time domain.
- 2 spatial coordinates, 1 time coordinate \mapsto 3DS-IVC Problem
- number of datapoints within one voxel is set as weight.¹
- Weight \propto computational costs.

¹Voxel is the 3D equivalent of a Pixel

Performance Profiles and Elapsed Time

(a) 2D Instances.¹(b) 3D Instances.¹

(c) ET for coloring solution 2D



(d) ET for coloring solution 2D

¹e.g. $\tau = 2$ and $\text{prop} = 0.8$ means algo is within twice the best-known for 80%.

-  D. Durrman and E. Saule, "Coloring the vertices of 9-pt and 27-pt stencils with intervals," in **2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)**, 2022, pp. 963–973. DOI: [10.1109/IPDPS53621.2022.00098](https://doi.org/10.1109/IPDPS53621.2022.00098).
-  A. Hohl, E. Delmelle, W. Tang, and I. Casas, "Accelerating the discovery of space-time patterns of infectious diseases using parallel computing," **Spatial and Spatio-temporal Epidemiology**, vol. 19, pp. 10–20, 2016, ISSN: 1877-5845. DOI: <https://doi.org/10.1016/j.sste.2016.05.002>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S187758451530040X>.
-  E. Saule, D. Panchanadam, A. Hohl, W. Tang, and E. M. Delmelle, "Parallel space-time kernel density estimation," **2017 46th International Conference on Parallel Processing (ICPP)**, pp. 483–492, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:6645797>.