TECHNISCHE
UNIVERSITÄT
WIEN

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 6, 2023

Outline

Connection between VCP and Parallel Algorithms

Connection between VCP and Parallel Algorithms

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

Introduction

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?
- $\mapsto$ 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?
- $\mapsto$ 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?
- $\mapsto$ 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.

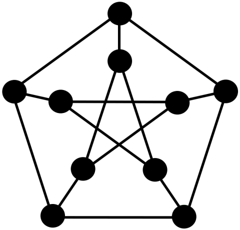
Connection between VCP and Parallel Algorithms

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

Vertex Color Problem Definition

- Given G(V,E)

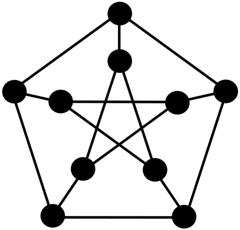
G(V,E)



Vertex Color Problem Definition

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

G(V,E)



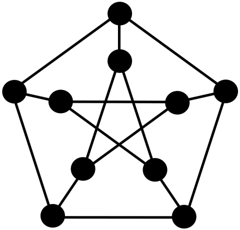
Vertex Color Problem Definition

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

Formal Definition of VCP given G(V,E)

find $f(v)$:
$\forall v \in V : \forall w \, in \, \Gamma(v) : f(v) \neq f(w).$

G(V,E)



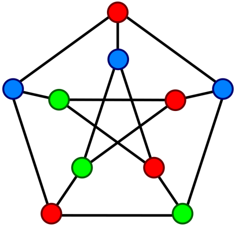
Vertex Color Problem Definition

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

Formal Definition of VCP given G(V,E)

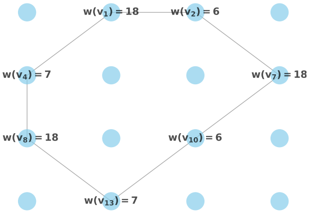
find $f(v)$:
$\forall v \in V : \forall w \, in \, \Gamma(v) : f(v) \neq f(w)$.

G(V,E)
Colored



Interval Vertex Coloring Definitions

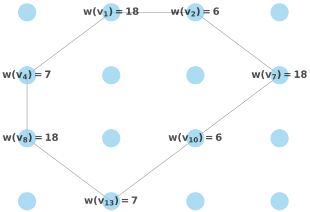
G(V,E)



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

Interval Vertex Coloring Definitions

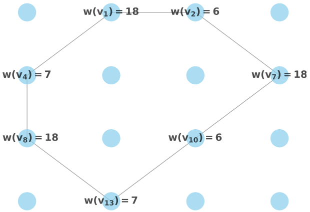
G(V,E)



- 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 start : $V \mapsto \mathbb{Z}^+$

Interval Vertex Coloring Definitions

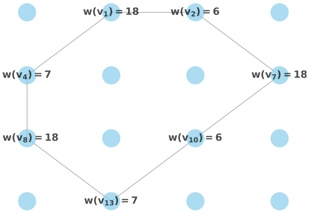
G(V,E)



- 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 start : $V \mapsto \mathbb{Z}^+$
- Vertex v is colored with open interval: $[\text{start}(v), \text{start}(v) + w(v))$

Interval Vertex Coloring Definitions

G(V,E)



- 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 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: $\forall (a, b) \in E$: $[\text{start}(a), \text{start}(a) + w(a)) \cap [\text{start}(b), \text{start}(b) + w(b)) = \emptyset$.

Formal IVC Problem Definition

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

Formal IVC Problem Definition

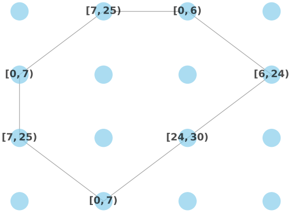
- A interval coloring of the vertices of G is a function start : $V \mapsto \mathbb{Z}^+$

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

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

Formal IVC Problem Definition

Interval Colored G(V,E)
with maxcolor = 30



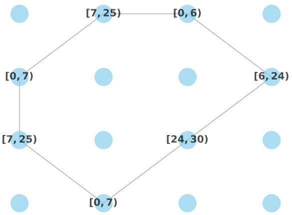
- A interval coloring of the vertices of G is a function start : $V \mapsto \mathbb{Z}^+$

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

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

Formal IVC Problem Definition

Interval Colored G(V,E)
with maxcolor = 30



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

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

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

Optimization Problem Instance:

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

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

Definitions for Analysis

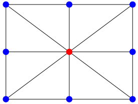
Definition: 2D Stencil Interval Vertex Coloring (2DS-IVC)

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 $|j - j'| \leq 1$.

Definitions for Analysis

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 $|j - j'| \leq 1$.



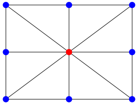
Definitions for Analysis

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 $|k - k'| \leq 1$.



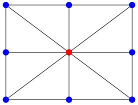
Definitions for Analysis

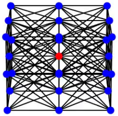
Definition: 2D Stencil Interval Vertex Coloring (2DS-IVC)

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 $|k - k'| \leq 1$.





Cliques K_n

Theorem

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

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

is indeed an optimal coloring maxcolor .*

Cliques K_n

Theorem

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

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

is indeed an optimal coloring maxcolor .*

- There exists such a coloring by listing vertices in descending order of weights and greedily allocating the color inteveral with the lowest available start(v) with $\mathcal{O}(n)$ $\square$.

Cliques K_n

Theorem

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

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

is indeed an optimal coloring maxcolor .*

- There exists such a coloring by listing vertices in descending order of weights and greedily allocating the color inteveral 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.

Cliques K_n

Theorem

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

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

is indeed an optimal coloring maxcolor .*

- There exists such a coloring by listing vertices in descending order of weights and greedily allocating the color inteveral 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.
- Therefore any identified K_4 in a 2D-IVC Problem and any K_8 in a 3D-IVC Problem are immediately (lose) lower bounds on maxcolor.

Bipartite Graphs

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.

Bipartite Graphs

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: maxcolor* $\geq w(i) + w(j), \forall (i,j) \in E$.

Bipartite Graphs

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:
  maxcolor* $\geq w(i) + w(j), \forall (i,j) \in E$.
- There exists a coloring
  maxcolor* $= \max_{(i,j) \in E} w(i) + w(j)$ in $\mathcal{O}(|E|)$. (Proof Omitted)

Bipartite Graphs

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: maxcolor* $\geq w(i) + w(j), \forall (i,j) \in E$.
- There exists a coloring 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 constains at least a Bipartite 7-pt stencil, and all paths are Bipartite.

Bipartite Graphs

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: maxcolor* $\geq w(i) + w(j), \forall (i,j) \in E$.
- There exists a coloring 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 constains at least a Bipartite 7-pt stencil, and all paths are Bipartite.
- Authors use this to construct approximation algorithms.

Odd Cycles 1

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

Odd Cycles 1

- 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.

Odd Cycles 1

- 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 maxcolor* $= 30$.

Odd Cycles 1

- 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 maxcolor* $= 30$.
- Understanding of coloring of odd cycles yields new lower bounds.

Odd Cycles 2

Definition

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

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

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

Odd Cycles 2

Definition

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

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

Definition

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

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

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

Odd Cycles 2

Definition

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

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

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:

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

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

Odd Cycles 3

Idea of Proof

Odd Cycles 3

Idea of Proof

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

Odd Cycles 3

Idea of Proof

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

Odd Cycles 3

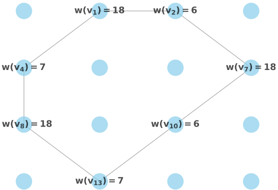
Idea of Proof

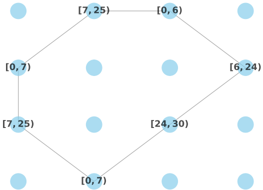
- If G is an odd cycle, there is an algorithm that yields max(maxpair, minchain3) colors, which means maxcolor* $\leq$ max(maxpair, minchain3).
- If G is an odd cycle, then maxcolor* $\geq$ max(maxpair, minchain3).
- Coinciding bounds conclude proof [1] $\square$.

Odd Cycles 3

Idea of Proof

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





3DS-IVC $\in$ NP ?

Lemma

3DS-IVC $\in$ *NP*

3DS-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.

3DS-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.
Verify that no adjacent edges have overlapping intervals.
$\forall (u, v) \in E$:

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

3DS-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.
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 $\mathcal{O}(n^2)$ $\square$.

NAE-3SAT $\propto$ 3DS-IVC ?

Idea of Proof

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

NAE-3SAT $\propto$ 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.

NAE-3SAT $\propto$ 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.

NAE-3SAT $\propto$ 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.
- 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 $\square$.

NAE-3SAT $\propto$ 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.
- 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 $\square$.

The proof yields:

Theorem

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

Algorithms: Bipartite Decomposition

Recall: Theorem

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

Algorithms: Bipartite Decomposition

Recall: Theorem

For bipartite graphs there exists a coloring
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 $\mathrm{RC} = \max c(x, y) + w(x, y)$ be the maximum color used by any of the rows.

Algorithms: Bipartite Decomposition

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 $\text{RC} = \max c(x, y) + w(x, y)$ be the maximum color used by any of the rows.

Corollary

$RC \leq maxcolor^*$ 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.

Algorithms: Bipartite Decomposition

```
Algorithm: Bipartite Decomposition
  Data: 2DS-IVC instance with Y rows
  Result: Approximate coloring with at most 2 · maxcolor*
for each row r in the 2DS-IVC instance do
    if r is even then
        Color the chain of vertices in row r optimally using colors from [0, RC);
    else
        Color the chain of vertices in row r optimally using colors from [RC, 2RC);
return Approximate coloring with at most 2 · maxcolor*
```

Algorithms: Bipartite Decomposition

```
Algorithm: Bipartite Decomposition
  Data: 2DS-IVC instance with Y rows
  Result: Approximate coloring with at most 2 · maxcolor*
for each row r in the 2DS-IVC instance do
    if r is even then
        Color the chain of vertices in row r optimally using colors from [0, RC);
    else
        Color the chain of vertices in row r optimally using colors from [RC, 2RC);
return Approximate coloring with at most 2 · maxcolor*
```

Theorem

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

Algorithms: Bipartite Decomposition

Algorithm: Bipartite Decomposition

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

```
for each row r in the 2DS-IVC instance do
    if r is even then
        Color the chain of vertices in row r optimally using colors from [0, RC);
    else
        Color the chain of vertices in row r optimally using colors from [RC, 2RC);
return Approximate coloring with at most 2 · maxcolor*
```

Theorem

Since the coloring obtained by the algorithm is at most $2 \cdot maxcolor^*$, 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].

Application and Experiments

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

Application and Experiments

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.

Application and Experiments

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

Application and Experiments

Space-Time Kernel Density Estimation

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

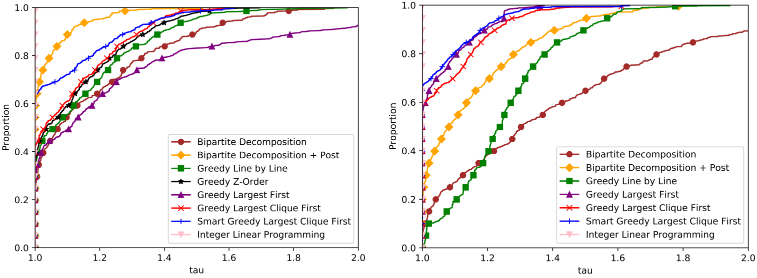
- Parallelized summation leads to race condition.
- Implementation of space-time decomposition by Hohl et. al. [2].
- Parallelization by exploiting symmetry by Saule et. al. [3].

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.[2]
- Weight $\propto$ computational costs.

[2] Voxel is the 3D equivalent of a Pixel

Results



(a) 2D Results

(b) 3D Results

Figure: Performance Profiles with ILP. [1]

References

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.

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. Panchananam, 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.