Deconstructing Intractability: A Case Study for **Interval Constrained Coloring**

Johannes Uhlmann

joint work with

Christian Komusiewicz and Rolf Niedermeier

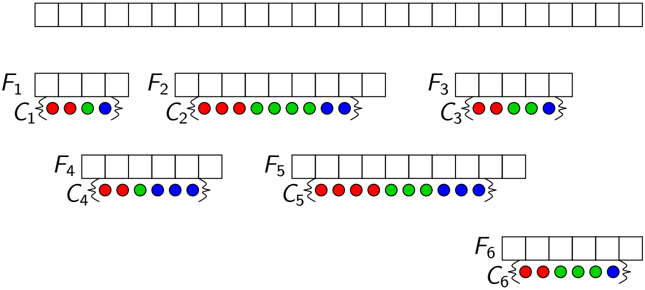
Friedrich-Schiller-Universität Jena
Institut für Informatik
Germany

CPM 2009, Lille

Outline

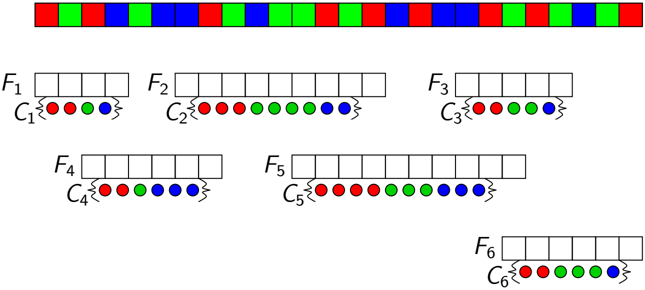
Interval Constrained Coloring (ICC)

Input:
integer intervals $\{F_1, \ldots, F_m\}$, $F_i \subseteq [1, n]$,
multisets of colors $\{C_1, \ldots, C_m\}$ over k colors.
Question:
Does there exists a coloring $c : [n] \to [k]$ such that $C_i = c(F_i)$, $\forall i$?



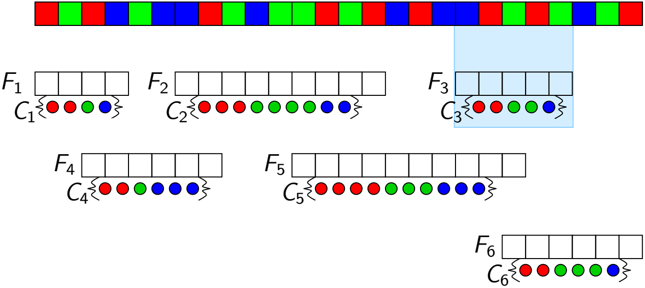
Interval Constrained Coloring (ICC)

Input:
integer intervals $\{F_1, \ldots, F_m\}$, $F_i \subseteq [1, n]$,
multisets of colors $\{C_1, \ldots, C_m\}$ over k colors.
Question:
Does there exists a coloring $c : [n] \to [k]$ such that $C_i = c(F_i)$, $\forall i$?



Interval Constrained Coloring (ICC)

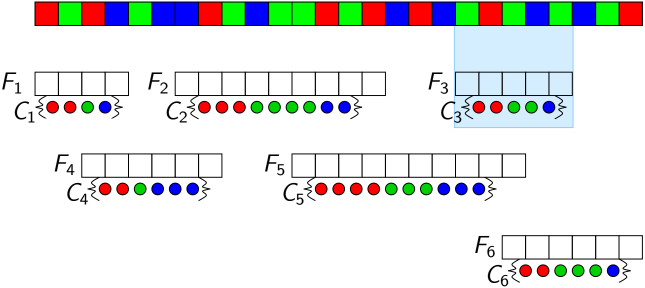
Input:
integer intervals $\{F_1, \ldots, F_m\}$, $F_i \subseteq [1, n]$,
multisets of colors $\{C_1, \ldots, C_m\}$ over k colors.
Question:
Does there exists a coloring $c : [n] \to [k]$ such that $C_i = c(F_i)$, $\forall i$?



C_3 is *matched* by coloring c.

Interval Constrained Coloring (ICC)

Input:
integer intervals $\{F_1, \ldots, F_m\}$, $F_i \subseteq [1, n]$,
multisets of colors $\{C_1, \ldots, C_m\}$ over k colors.
Question:
Does there exists a coloring $c : [n] \to [k]$ such that $C_i = c(F_i)$, $\forall i$?



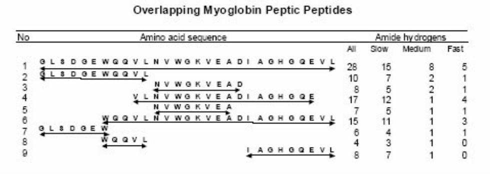
C_3 is NOT matched.

Motivation

Application: interpretation of experimental data (mass spectrometry).

Goal: Obtain information about 3D structure of a protein.

- intervals: protein fragments (amino acid sequences)
- colors: Hydrogen/Deuterium exchange rates (Slow, Medium, Fast)
- ICC:
  assign exchange rates to single amino acids



Overlapping Myoglobin Peptic Peptides

No	Amino acid sequence	Amide hydrogens All	Slow	Medium	Fast
1	G L S D G E W Q Q V L N V W G K V E A D I A G H G Q E V L	28	15	8	5
2	G L S D G E W Q Q V L	10	7	2	1
3	N V W G K V E A D	8	5	2	1
4	V L N V W G K V E A D I A G H G Q E	17	12	1	4
5	N V W G K V E A	7	5	1	1
6	W Q Q V L N V W G K V E A D I A G H G Q E V L	15	11	1	3
7	G L S D G E W	6	4	1	1
8	W Q Q V L	4	3	1	0
9	I A G H G Q E V L	8	7	1	0

[Althaus, Canzar, Emmett, Karrenbauer, Marshall, Meyer-Baese, and Zhang, ACM Symposium on Applied Computing (SAC 08)]

Previous work

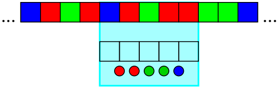
Two recent papers:

[Althaus, Canzar, Emmett, Karrenbauer, Marshall, Meyer-Baese, and Zhang, ACM Symposium on Applied Computing (SAC 08)]

- ILP and branch-and-bound based algorithm
- polynomial-time solvable for two colors

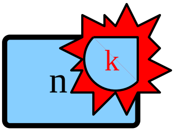
[Althaus, Canzar, Elbassioni, Karrenbauer, and Mestre, 11th Scandinavian Workshop on Algorithm Theory (SWAT 08)]

- NP-complete
- approximation algorithm: every color multi-set is matched within ± 1 for every color

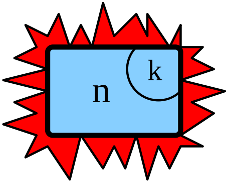


Parameterized Complexity

Given an NP-hard problem with input size n and a parameter k
Basic idea: Confine the combinatorial explosion to k



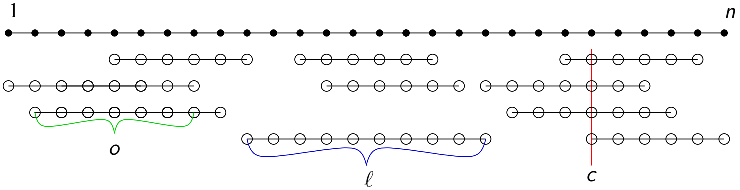
instead of



Definition

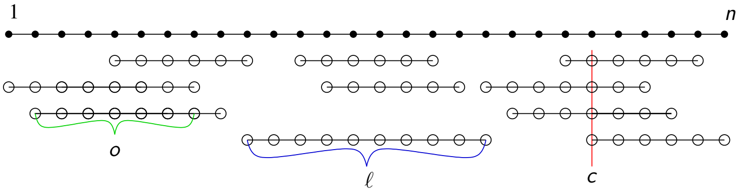
A problem of size n is called *fixed-parameter tractable* with respect to a parameter k if it can be solved in $f(k) \cdot n^{O(1)}$ time.

Parameter Identification



n	range
m	number of intervals
k	number of colors
ℓ	maximum interval length
c	cutwidth $\max_{i \in [1,n]} \lvert\{F_j \mid i \in F_j\}\rvert$
o	maximum overlap between intervals

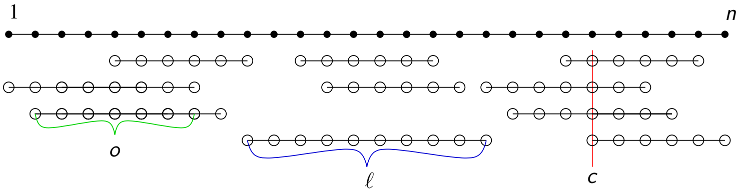
Parameter Identification



ICC is NP-hard for cutwidth $c = 3$ [Althaus et al., SWAT 08].
All other parameters are non-constant in this NP-hardness proof.

- What is the parameterized complexity with respect to these non-constant parameters?
- Can ICC be solved in polynomial time for cutwidth two?

Parameter Identification



ICC is NP-hard for cutwidth $c = 3$ [Althaus et al., SWAT 08].
All other parameters are non-constant in this NP-hardness proof.

- What is the parameterized complexity with respect to these non-constant parameters?
- Can ICC be solved in polynomial time for cutwidth two?

What parameter assumptions does the NP-hardness proof make?
What is the complexity of the problem if these assumptions do not hold?

Overview

Single parameters

Parameter	k	ℓ	c	m	n	o
Complexity	$k = 2$: P $k \geq 3$: ?	$\ell!$	$c = 2$: P $c = 3$: NPc	?	$n!$	$o = 1$: P $o \geq 2$: ?

k: number of colors, ℓ: max. interval length, c: cutwidth, m: number of intervals, n: input range, o: max. overlap

Althaus et al. (2 papers)

Overview

Single parameters

Parameter	k	ℓ	c	m	n	o
Complexity	$k = 2$: P $k \geq 3$: ?	$\ell!$	$c = 2$: P $c = 3$: NPc	?	$n!$	$o = 1$: P $o \geq 2$: ?

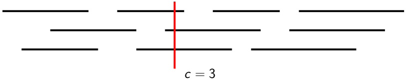
Combined parameters

Parameter	$(k, *)$	$(k, *, *)$	$(\ell, *)$
Running times	k^ℓ $(k-1)^n$ $f(k, m)$ (ILP)	$\ell^{c \cdot (k-1)}$, $n^{c \cdot (k-1)}$	$(c+1)^\ell$

k: number of colors, ℓ: max. interval length, c: cutwidth,
m: number of intervals, n: input range, o: max. overlap

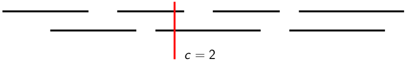
Althaus et al. (2 papers)

Cutwidth



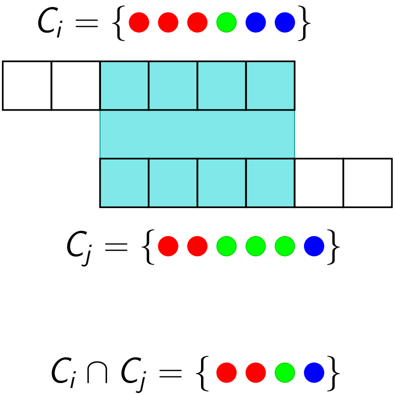
ICC is NP-complete for cutwidth $c = 3$.

[Althaus, Canzar, Elbassioni, Karrenbauer, and Mestre, SWAT 08]



ICC can be solved in $O(n^2)$ time for cutwidth $c = 2$.

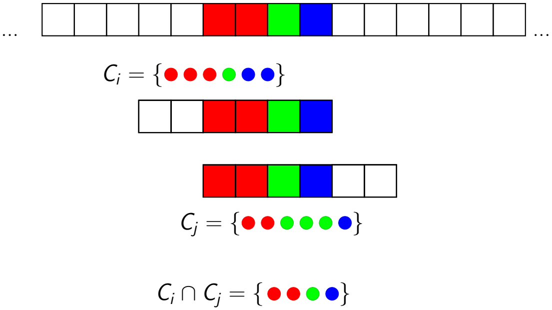
Solving strategy: polynomial-time data reduction rules.



Rule 1

For any two intervals F_i and F_j,

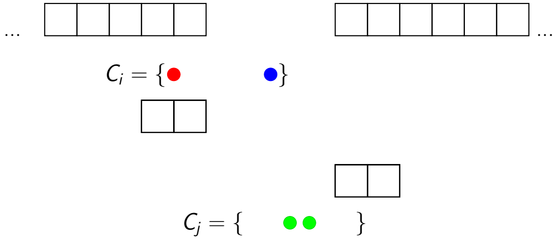
- if $|F_i \cap F_j| = |C_i \cap C_j|$, then $c(F_i \cap F_j) := C_i \cap C_j$;
- if $|F_i \cap F_j| > |C_i \cap C_j|$, then return "No".



Rule 1

For any two intervals F_i and F_j,

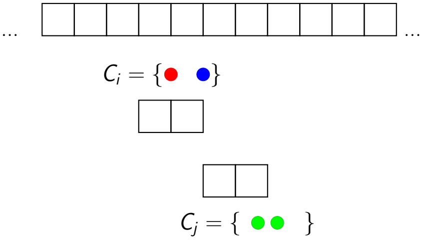
- if $|F_i \cap F_j| = |C_i \cap C_j|$, then $c(F_i \cap F_j) := C_i \cap C_j$;
- if $|F_i \cap F_j| > |C_i \cap C_j|$, then return “No”.



Rule 1

For any two intervals F_i and F_j,

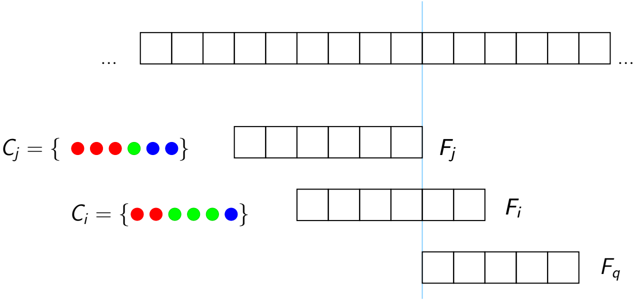
- if $|F_i \cap F_j| = |C_i \cap C_j|$, then $c(F_i \cap F_j) := C_i \cap C_j$;
- if $|F_i \cap F_j| > |C_i \cap C_j|$, then return “No”.



Rule 1

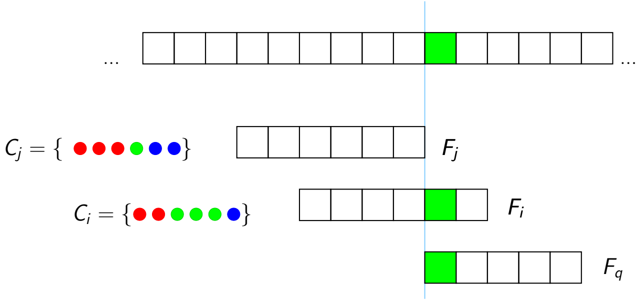
For any two intervals F_i and F_j,

- if $|F_i \cap F_j| = |C_i \cap C_j|$, then $c(F_i \cap F_j) := C_i \cap C_j$;
- if $|F_i \cap F_j| > |C_i \cap C_j|$, then return "No".



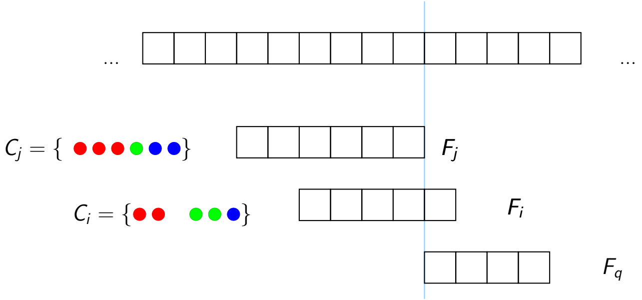
Rule 2

F_i: interval whose positions are all contained in further intervals.
F_j: interval overlapping with F_i.
x: color that occurs more often in C_i than in C_j.
If $x \in C_q$
  $c(p) := x$ for a $p \in F_i \setminus F_j$,
else
  "no".



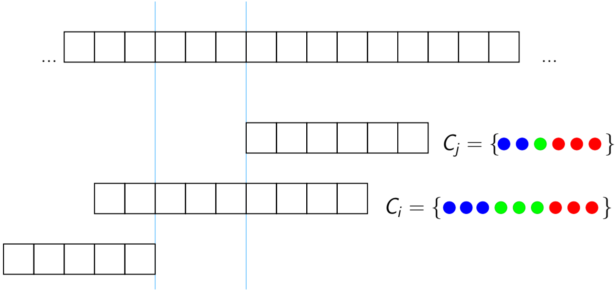
Rule 2

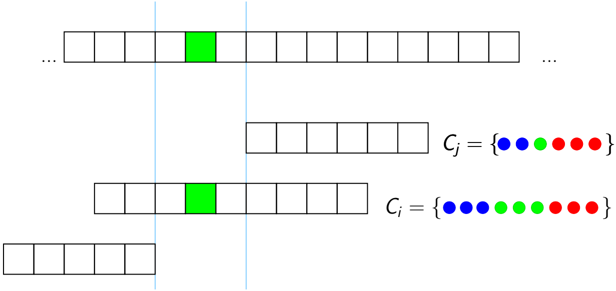
F_i: interval whose positions are all contained in further intervals.
F_j: interval overlapping with F_i.
x: color that occurs more often in C_i than in C_j.
If $x \in C_q$
 $c(p) := x$ for a $p \in F_i \setminus F_j$,
else
 "no".

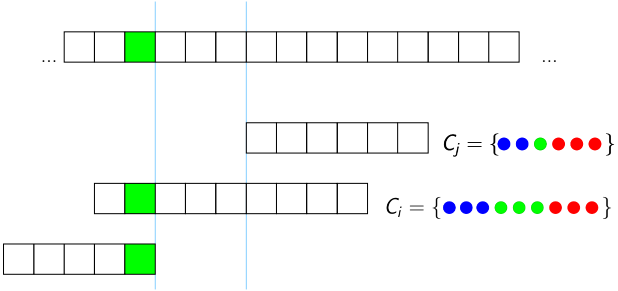


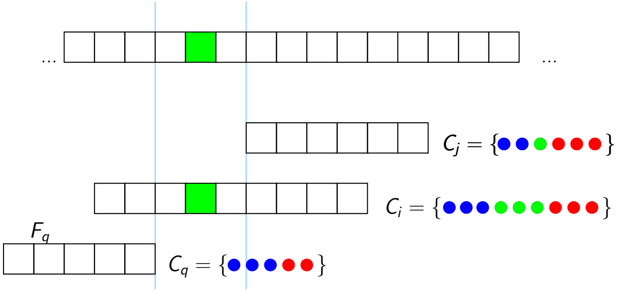
Rule 2

F_i: interval whose positions are all contained in further intervals.
F_j: interval overlapping with F_i.
x: color that occurs more often in C_i than in C_j.
If $x \in C_q$
 $c(p) := x$ for a $p \in F_i \setminus F_j$,
else
 "no".


...
...
C_j =
C_i =

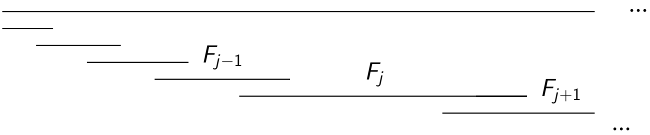

...
...
$C_j = \{$ $\}$
$C_i = \{$ $\}$


...
...
$C_j = \{ \}$
$C_i = \{ \}$



Rule 3.1

F_i, F_j: overlapping intervals.
F_q: another interval overlapping with F_i.
x: color that occurs more often in C_i than in C_j.
If $x \notin C_q$, then $c(p) := x$ for a $p \in F_i \setminus (F_j \cup F_q)$



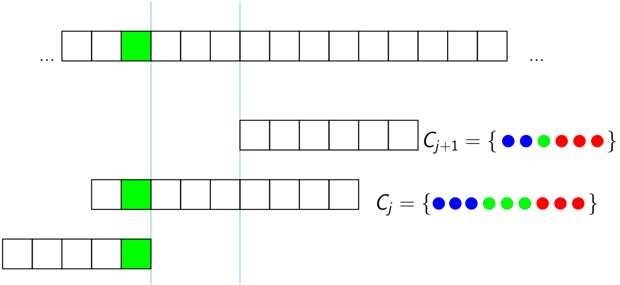
Rule 3.2

Consider a reduced instance.
Let the input intervals be sorted by their starting points.
Let j be minimum number such that

- exists a color $x \in C_j \cap C_{j-1}$ that occurs more often in C_j than in C_{j+1}

Then, set $c(p) = x$ for some $p \in F_j \cap F_{j-1}$

$C_1 \subseteq C_2 \subseteq \ldots \subseteq C_j$



Rule 3.2

Consider a reduced instance.
Let the input intervals be sorted by their starting points.
Let j be minimum number such that

- exists a color $x \in C_j \cap C_{j-1}$ that occurs more often in C_j than in C_{j+1}

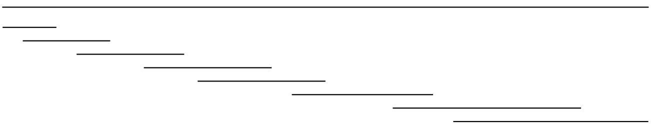
Then, set $c(p) = x$ for some $p \in F_j \cap F_{j-1}$

$C_1 \subseteq C_2 \subseteq \ldots \subseteq C_j$

Cutwidth

Apply data reduction rules exhaustively.

- Either output "no"
- or reduced instance with $C_1 \subseteq C_2 \subseteq \ldots \subseteq C_m$



Theorem

ICC can be solved in $O(n^2)$ time for cutwidth $c = 2$.

Overview

Single parameters

Parameter	k	ℓ	c	m	n	o
Complexity	$k = 2$: P $k \geq 3$: ?	$\ell!$	$c = 2$: P $c = 3$: NPc	?	$n!$	$o = 1$: P $o \geq 2$: ?

k: number of colors, ℓ: max. interval length, c: cutwidth,
m: number of intervals, n: input range, o: max. overlap

Althaus et al. (2 papers)

Overview

Single parameters

Parameter	k	ℓ	c	m	n	o
Complexity	$k = 2$: P $k \geq 3$: ?	$\ell!$	$c = 2$: P $c = 3$: NPc	?	$n!$	$o = 1$: P $o \geq 2$: ?

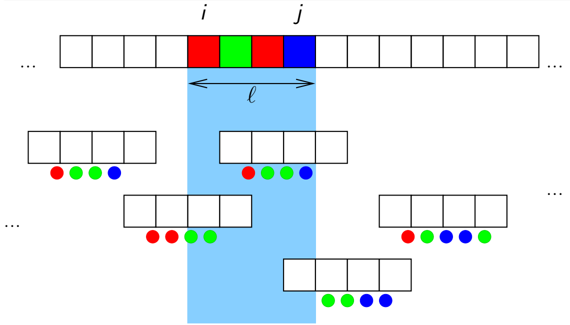
Combined parameters

Parameter	$(k, *)$	$(k, *, *)$	$(\ell, *)$
Running times	k^ℓ $(k-1)^n$ $f(k, m)$ (ILP)	$\ell^{c\cdot(k-1)}$, $n^{c\cdot(k-1)}$	$(c+1)^\ell$

k: number of colors, ℓ: max. interval length, c: cutwidth,
m: number of intervals, n: input range, o: max. overlap

Althaus et al. (2 papers)

ICC can be solved in $O(k^\ell(k+\ell m)n)$ time.

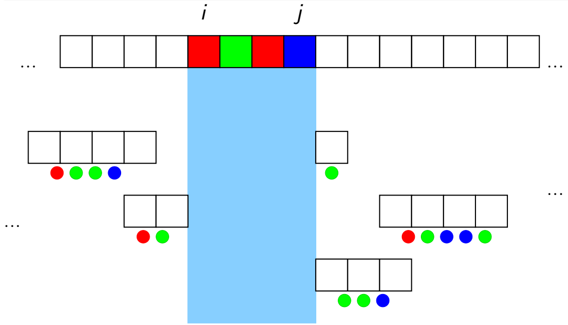


If the coloring of $[i,j]$ is known, then the instance restricted to $[j+1,n]$ can be solved independently from $[1,i-1]$.

Dynamic Programming

- traverse the input from left to right.
- for every length-ℓ subinterval maintain a table with an entry for every coloring.

ICC can be solved in $O(k^{\ell}(k+\ell m)n)$ time.

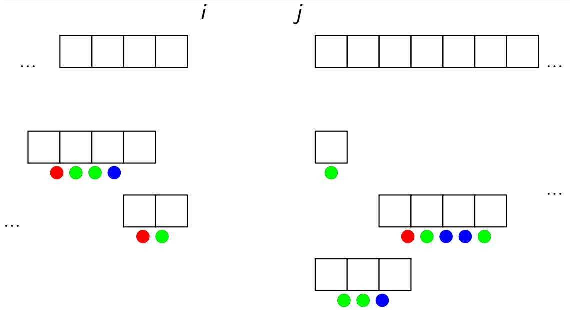


If the coloring of $[i,j]$ is known, then the instance restricted to $[j+1,n]$ can be solved independently from $[1,i-1]$.

Dynamic Programming

- traverse the input from left to right.
- for every length-ℓ subinterval maintain a table with an entry for every coloring.

ICC can be solved in $O(k^{\ell}(k+\ell m)n)$ time.



If the coloring of $[i, j]$ is known, then the instance restricted to $[j+1, n]$ can be solved independently from $[1, i-1]$.

Dynamic Programming

- traverse the input from left to right.
- for every length-ℓ subinterval maintain a table with an entry for every coloring.

Running time:

ICC can be solved in $O(k^\ell(k+\ell m)n)$ time.

With a similar algorithm we obtain:

ICC can be solved in $O(\ell! \cdot \ell mn)$ time.

Based on a second possibility to divide the input, we can show

ICC can be solved in

- $(c+1)^\ell \cdot poly(n,m)$ time, or
- $\ell^{c(k-1)} \cdot poly(n,m)$ time.

Open Questions

Single parameters

Parameter	k	ℓ	c	m	n	o
Complexity	$k = 2$: P $k \geq 3$: ?	$\ell!$	$c = 2$: P $c = 3$: NPc	?	$n!$	$o = 1$: P $o \geq 2$: ?

Combined parameters

Parameter	$(k, *)$	$(k, *, *)$	$(\ell, *)$
Running times	k^ℓ $(k-1)^n$ $f(k, m)$ (ILP)	$\ell^{c \cdot (k-1)}$, $n^{c \cdot (k-1)}$	$(c+1)^\ell$

- What is the complexity of ICC for $k = 3$ colors?
- Is ICC fixed-parameter tractable for the combined parameter (c, k)?
- Is ICC fixed-parameter tractable w.r.t. m the number of intervals?

k: number of colors, ℓ: max. interval length, c: cutwidth,
m: number of intervals, n: input range, o: max. overlap
Althaus et al. (2 papers)

Future Research

- Extend the investigations to the optimaztion variants.
- In the biochemical application $k = 3$.
  - We can show that ICC can be solved in $1.87^n \cdot poly(n, m)$ time for $k = 3$.
  - Can some of our other results be improved for the case $k = 3$?
- Implementation of our algorithms.

Thank you!