Online problems	Advice complexity	Paging	DiffServ	Conclusion
Но	w much informa ne	ition about eded ?	the future i	s
Stef	an Dobrev ¹ , <i>Rastisla</i>	av Královič ² , D)ana Pardubská	l ²
	Slovak Aca	of Mathematics, ademy of Sciences obrev@savba.sk		
		of Computer Scienc rsity, Bratislava, Slo		

{kralovic,pardubska}@dcs.fmph.uniba.sk

Sofsem 2008, January 21, 2008

ヨト イヨト

- 2

→ E > < E >

< 🗇 ▶

э

ski-rental problem

dilemma of a skier:

- rent equipment for 10 EUR per day
- buy equipment for 10c EUR

doesn't know how many days will be skiing

<ロ> (四) (四) (三) (三) (三) (三)

ski-rental problem

dilemma of a skier:

- rent equipment for 10 EUR per day
- buy equipment for 10c EUR

doesn't know how many days will be skiing

online algorithm $\ensuremath{\mathcal{A}}$

input $\mathbf{x} = \langle x_1, x_2, \dots, x_n \rangle$ output $\mathbf{y} = \mathcal{A}(\mathbf{x}) = \langle y_1, y_2, \dots, y_n \rangle$, where $y_i = f(x_1, \dots, x_i)$.

<ロ> <問> <問> < 回> < 回> < 回> < 回> < 回

ski-rental problem

dilemma of a skier:

- rent equipment for 10 EUR per day
- buy equipment for 10c EUR

doesn't know how many days will be skiing

online algorithm $\ensuremath{\mathcal{A}}$

input $\mathbf{x} = \langle x_1, x_2, \dots, x_n \rangle$ output $\mathbf{y} = \mathcal{A}(\mathbf{x}) = \langle y_1, y_2, \dots, y_n \rangle$, where $y_i = f(x_1, \dots, x_i)$.

c-competitive algorithm

for each input \boldsymbol{x} : $cost(\mathcal{A}(\boldsymbol{x})) \leq c \cdot cost(OPT(\boldsymbol{x}))$

・ロト ・聞 ト ・ ヨト ・ ヨト … ヨ

ski-rental problem

dilemma of a skier:

- rent equipment for 10 EUR per day
- buy equipment for 10c EUR

doesn't know how many days will be skiing

online algorithm $\ensuremath{\mathcal{A}}$

input $\mathbf{x} = \langle x_1, x_2, \dots, x_n \rangle$ output $\mathbf{y} = \mathcal{A}(\mathbf{x}) = \langle y_1, y_2, \dots, y_n \rangle$, where $y_i = f(x_1, \dots, x_i)$.

c-competitive algorithm

for each input \boldsymbol{x} : $cost(\mathcal{A}(\boldsymbol{x})) \leq c \cdot cost(OPT(\boldsymbol{x})) \Leftarrow offline$

イロト イポト イヨト イヨト

э

ski-rental problem

dilemma of a skier:

- rent equipment for 10 EUR per day
- buy equipment for 10c EUR

doesn't know how many days will be skiing

Theorem [Karp 1992]

optimal worst-case strategy: rent c - 1 days, then buy competitive ratio:

$$\frac{\operatorname{cost}(\mathcal{A}(\boldsymbol{x}))}{\operatorname{cost}(OPT(\boldsymbol{x}))} \leq \frac{2c-1}{c} = 2 - \frac{1}{c}$$

c-competitive algorithm

for each input \boldsymbol{x} : $cost(\mathcal{A}(\boldsymbol{x})) \leq c \cdot cost(OPT(\boldsymbol{x}))$

Definition: problem complexity

Best attainable competitive ratio.

- loose competitiveness,...: taylored for Paging
- resource augmentation: OPT vs. k-times "more powerfull" online
- Iook-ahead: alg. can see a number of future inputs
- online vs online: Max/Max ratio, relative worst-order ratio,...
- limited adversary: access graph, statistical, diffuse, ...
- entropy

.....

c-competitive algorithm

for each input \boldsymbol{x} : $cost(\mathcal{A}(\boldsymbol{x})) \leq c \cdot cost(OPT(\boldsymbol{x}))$

Definition: problem complexity

Best attainable competitive ratio.

- loose competitiveness,...: taylored for Paging
- resource augmentation: OPT vs. k-times "more powerfull" online
- look-ahead: alg. can see a number of future inputs
- online vs online: Max/Max ratio, relative worst-order ratio,...
- Iimited adversary: access graph, statistical, diffuse, ...
- entropy

c-competitive algorithm

for each input \boldsymbol{x} : $cost(\mathcal{A}(\boldsymbol{x})) \leq c \cdot cost(OPT(\boldsymbol{x}))$

Definition: problem complexity

Best attainable competitive ratio.

- loose competitiveness,...: taylored for Paging
- resource augmentation: OPT vs. k-times "more powerfull" online
- look-ahead: alg. can see a number of future inputs
- online vs online: Max/Max ratio, relative worst-order ratio,...
- Iimited adversary: access graph, statistical, diffuse, ...
- entropy

(人間) アイヨア イヨア

c-competitive algorithm

for each input \boldsymbol{x} : $cost(\mathcal{A}(\boldsymbol{x})) \leq c \cdot cost(OPT(\boldsymbol{x}))$

Definition: problem complexity

Best attainable competitive ratio.

- loose competitiveness,...: taylored for Paging
- resource augmentation: OPT vs. k-times "more powerfull" online
- look-ahead: alg. can see a number of future inputs
- online vs online: Max/Max ratio, relative worst-order ratio,...
- Iimited adversary: access graph, statistical, diffuse, ...
- entropy

c-competitive algorithm

for each input \boldsymbol{x} : $cost(\mathcal{A}(\boldsymbol{x})) \leq c \cdot cost(OPT(\boldsymbol{x}))$

Definition: problem complexity

Best attainable competitive ratio.

- loose competitiveness,...: taylored for Paging
- resource augmentation: OPT vs. k-times "more powerfull" online
- look-ahead: alg. can see a number of future inputs
- online vs online: Max/Max ratio, relative worst-order ratio,...
- limited adversary: access graph, statistical, diffuse, ...

< = >

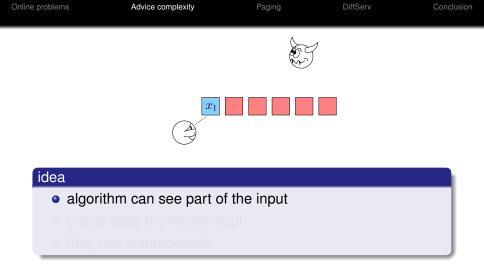
c-competitive algorithm

for each input \boldsymbol{x} : $cost(\mathcal{A}(\boldsymbol{x})) \leq c \cdot cost(OPT(\boldsymbol{x}))$

Definition: problem complexity

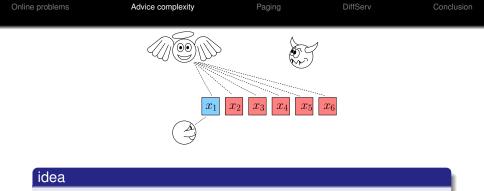
Best attainable competitive ratio.

- loose competitiveness,...: taylored for Paging
- resource augmentation: OPT vs. k-times "more powerfull" online
- look-ahead: alg. can see a number of future inputs
- online vs online: Max/Max ratio, relative worst-order ratio,...
- limited adversary: access graph, statistical, diffuse, ...
- entropy



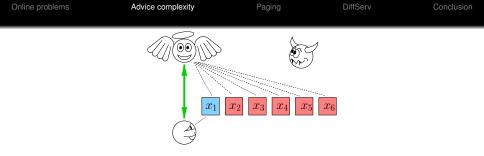
・ロト ・ 理 ト ・ ヨ ト ・

-



- algorithm can see part of the input
- oracle sees the whole input
- they can communicate

・ロト ・ 同ト ・ ヨト ・ ヨト … ヨ



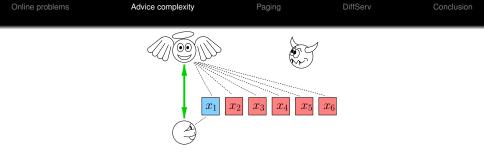
idea

- algorithm can see part of the input
- oracle sees the whole input
- they can communicate

→ Ξ → < Ξ →</p>

< < >> < </>

э



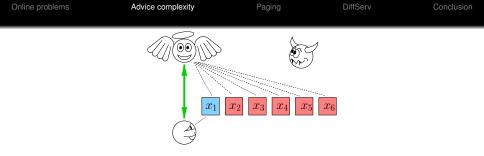
idea

- algorithm can see part of the input
- oracle sees the whole input
- they can communicate

problem complexity \approx # bits to achieve <code>OPT</code>

イロト イポト イヨト イヨト

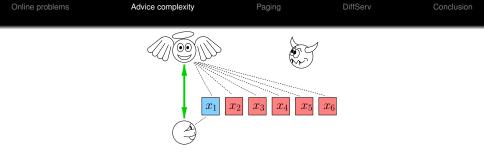
э



communication

- answerer: algorithm asks, gets an answer
- helper: oracle can give spontanuous advice

프 🕨 🗉 프



communication

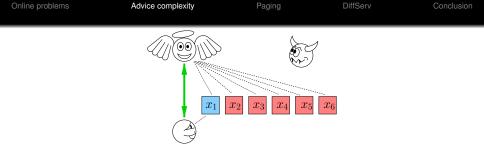
- answerer: algorithm asks, gets an answer
- helper: oracle can give spontanuous advice

trivial bounds

- encode whole input
- encode output

프 🖌 🛪 프 🕨

э



Definition: helper

input $\mathbf{x} = \langle x_1, x_2, \dots, x_n \rangle$ helper sequence: $\mathcal{O}(\mathbf{x}) = \langle \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n \rangle$ output: $\mathbf{y} = \langle y_1, y_2, \dots, y_n \rangle$, $y_i = f(x_1, \dots, x_i, \mathbf{a}_1, \dots, \mathbf{a}_i)$

advice (bit) complexity:

$$B_{(\mathcal{A},\mathcal{O})}^{H} = \limsup_{n \mapsto \infty} \max_{|\mathbf{X}| = n} \frac{\sum_{i=1}^{n} |\mathbf{a}_i|}{n}$$

・ 同 ト ・ 三 ト ・

Online problems	Advice complexity	Paging	DiffServ	Conclusion
our results				

	competitive ratio	helper	answerer	
PAGING	К	(0.1775, 0.2056)	$(0.4591, 0.5 + \varepsilon)$	
DIFFSERV	≈ 1.281	$\frac{1}{K}$	$\left(\frac{\log K}{2K}, \frac{\log K}{K}\right)$	





- input: logical pages $\boldsymbol{x} = \langle x_1, x_2, \dots, x_n \rangle, x_i > 0$
- buffer: physical memory $B = \{b_1, \dots, b_K\}$

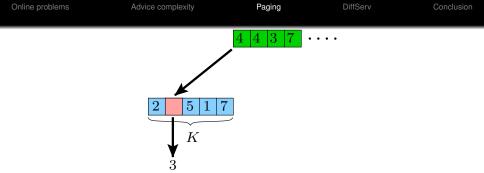
• If $x_i \notin B \Rightarrow$ page fault, a victim has to be four

Theorem [Sleator, Tarjan 1985]

Every deterministic algorithm is at least K competitive.

Stefan Dobrev, Rastislav Královič, Dana Pardubská How much information about the future is needed ?

・ 言 ・ ・ ヨ ・ ・ 雪 ・



- input: logical pages $\boldsymbol{x} = \langle x_1, x_2, \dots, x_n \rangle$, $x_i > 0$
- buffer: physical memory $B = \{b_1, \ldots, b_K\}$
- if $x_i \notin B \Rightarrow$ page fault, a victim has to be found

Theorem [Sleator, Tarjan 1985]

Every deterministic algorithm is at least K competitive.

Stefan Dobrev, Rastislav Královič, Dana Pardubská How much information about the future is needed ?

白マト イヨト イヨト

• input: logical pages $\mathbf{x} = \langle x_1, x_2, \dots, x_n \rangle$, $x_i > 0$ • buffer: physical memory $B = \{b_1, \dots, b_K\}$ • if $x_i \notin B \Rightarrow$ page fault, a victim has to be found	Paging				
\checkmark If $X_1 \not\subset D \rightarrow D$ age fault, a victim has to be found	input:buffer	: physical memory	$B = \{b_1, \ldots, b_n\}$	b_K	

online algorithm with helper

1 bit per input request ightarrow can be optimized

Stefan Dobrev, Rastislav Královič, Dana Pardubská How much information about the future is needed ?

N H R N H R R N H R R N H R R N

÷

Online	e problems	Advice complexity	Paging	DiffServ	Conclusion	
	Paging					
	input: log	gical pages $\pmb{x}=$	$\langle x_1, x_2, \ldots, x_n$	$\rangle, x_i > 0$		
	buffer: p	hysical memory	$B = \{b_1, \ldots, b_n\}$	b _K }		
	If x _i ∉ B	\Rightarrow page fault, a v	victim has to I	be found		

Replace the farthest-requested page.

online algorithm with helper

1 bit per input request ightarrow can be optimized

Stefan Dobrev, Rastislav Královič, Dana Pardubská How much information about the future is needed ?

-

Onlin	e problems	Advice complexity	Paging	DiffServ	Conclusion	
	Paging					
	input: log	gical pages $\pmb{x}=$	$\langle x_1, x_2, \ldots, x_n$	$\rangle, x_i > 0$		
	buffer: p	hysical memory	$B = \{b_1, \ldots, b_n\}$	<i>b</i> _К }		
	• if $x_i \notin B$	\Rightarrow page fault, a v	victim has to I	oe found		

Replace the farthest-requested page.

online algorithm with helper

- a page is active, if shall be used by OPT
- replace only inactive pages
- with each input helper tells if the page is active

1 bit per input request ightarrow can be optimized

0	nline problems	Advice complexity	Paging	DiffServ	Conclusion	
	Paging					
	input:	logical pages $\boldsymbol{x} =$	$\langle x_1, x_2, \ldots, x_l \rangle$	$ x_i\rangle, x_i>0$		
	buffer	: physical memory	$B = \{b_1, \ldots, $	b_K }		
	 if x_i ∉ 	$B \Rightarrow$ page fault, a	victim has to	be found		

Replace the farthest-requested page.

online algorithm with helper

- a page is active, if shall be used by OPT
- replace only inactive pages

with each input helper tells if the page is active

1 bit per input request \rightarrow can be optimized

DER NERNER E

Onli	ine problems	Advice complexity	Paging	DiffServ	Conclusi	ion
	Paging					
	input: lo	ogical pages x =	$\langle x_1, x_2, \ldots, x_n \rangle$	$\langle x_i \rangle, x_i > 0$		
• buffer: physical memory $B = \{b_1, \dots, b_K\}$						
	● if <i>x_i ∉ E</i>	$B \Rightarrow$ page fault, a v	victim has to	be found		

Replace the farthest-requested page.

online algorithm with helper

- a page is active, if shall be used by OPT
- replace only inactive pages
- with each input helper tells if the page is active

1 bit per input request \rightarrow can be optimized

Onli	ine problems	Advice complexity	Paging	DiffServ	Conclusi	ion
	Paging					
	input: lo	ogical pages x =	$\langle x_1, x_2, \ldots, x_n \rangle$	$\langle x_i \rangle, x_i > 0$		
• buffer: physical memory $B = \{b_1, \dots, b_K\}$						
	● if <i>x_i ∉ E</i>	$B \Rightarrow$ page fault, a v	victim has to	be found		

Replace the farthest-requested page.

online algorithm with helper

- a page is active, if shall be used by OPT
- replace only inactive pages
- with each input helper tells if the page is active

1 bit per input request \rightarrow can be optimized

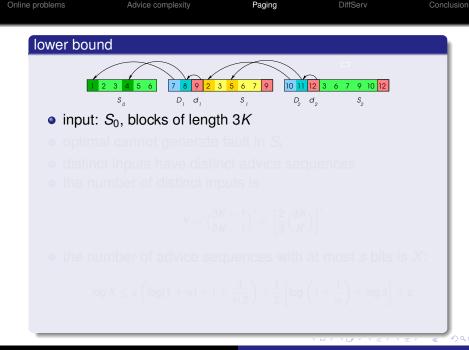
Online problems	Advice complexity	Paging	DiffServ	Conclusion
Paging				
input:	logical pages $\boldsymbol{x} =$	$\langle x_1, x_2, \ldots, x_r \rangle$	$\langle x_i \rangle, x_i > 0$	
buffer	: physical memory	$B = \{b_1, \ldots, $	b_K }	
● if <i>x_i ∉</i>	$B \Rightarrow$ page fault, a	victim has to	be found	

Replace the farthest-requested page.

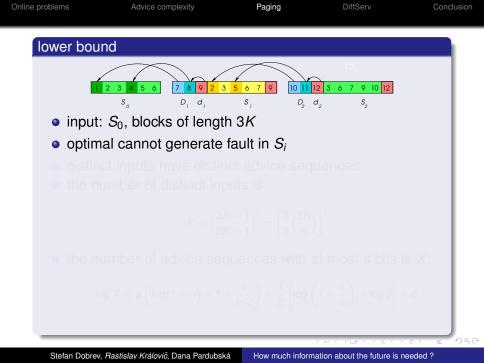
online algorithm with helper

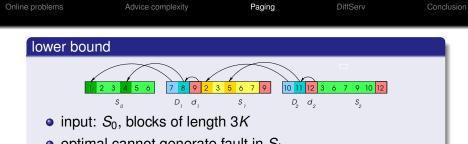
- a page is active, if shall be used by OPT
- replace only inactive pages
- with each input helper tells if the page is active

1 bit per input request \rightarrow can be optimized



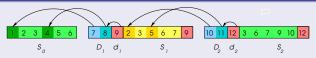
Stefan Dobrev, Rastislav Královič, Dana Pardubská How much information about the future is needed ?





- optimal cannot generate fault in S_i
- distinct inputs have distinct advice sequences

lower bound



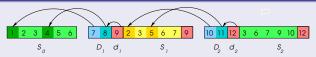
- input: S₀, blocks of length 3K
- optimal cannot generate fault in S_i
- distinct inputs have distinct advice sequences
- the number of distinct inputs is

$$Y = \binom{3K-1}{2K-1}^{i} = \left[\frac{2}{3}\binom{3K}{K}\right]^{i}$$

• the number of advice sequences with at most *s* bits is *X*: $\log X \le s \left(\log(1+\alpha) + 1 + \frac{1}{\ln 2} \right) + \frac{1}{2} \left[\log \left(1 + \frac{1}{\alpha} \right) + \log s \right] + c$

How much information about the future is needed ?

lower bound

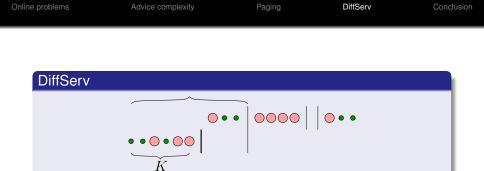


- input: S₀, blocks of length 3K
- optimal cannot generate fault in S_i
- distinct inputs have distinct advice sequences
- the number of distinct inputs is

$$Y = {\binom{3K-1}{2K-1}}^i = \left[\frac{2}{3} {\binom{3K}{K}}\right]^i$$

• the number of advice sequences with at most *s* bits is *X*:

$$\log X \leq s \left(\log(1+\alpha) + 1 + \frac{1}{\ln 2} \right) + \frac{1}{2} \left[\log \left(1 + \frac{1}{\alpha} \right) + \log s \right] + c$$

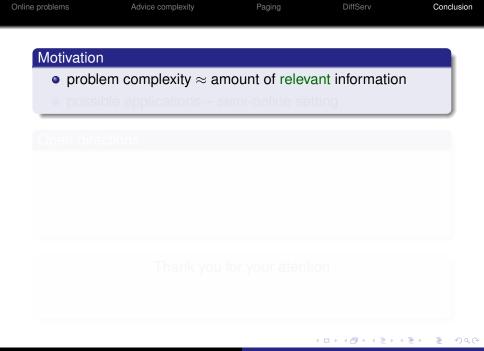


- buffer of size K, input: sequence of large and small items
- process one item, discard some to fit into buffer
- order must be preserved

skipped

▲ 圖 ▶ ▲ 国 ▶ ▲ 国 ▶ ……

3



Online problems	Advice complexity	Paging	DiffServ	Conclusion
Mativation				
Motivation		nount of rolov	ant information	
	em complexity $pprox$ ar ole applications – s			
• possi	sie applications – s	enn-onime se	ling	

Stefan Dobrev, Rastislav Královič, Dana Pardubská How much information about the future is needed ?

Online problems	Advice complexity	Paging	DiffServ	Conclusion

- problem complexity pprox amount of relevant information
- possible applications semi-online setting

Open directions

- other problems
- bounded advice
- trade-off between approximation and information
- randomization

Thank you for your atention

・ロト ・ 同ト ・ ヨト ・ ヨト … ヨ

Online problems	Advice complexity	Paging	DiffServ	Conclusion

- problem complexity pprox amount of relevant information
- possible applications semi-online setting

Open directions

- other problems
- bounded advice
- trade-off between approximation and information.
- randomization

Thank you for your atention

・ロン・(部)とくほどくほどう ほ

Online problems	Advice complexity	Paging	DiffServ	Conclusion

- problem complexity \approx amount of relevant information
- possible applications semi-online setting

Open directions

- other problems
- bounded advice
- trade-off between approximation and information

randomization

Thank you for your atention

ヘロト ヘアト ヘビト ヘビト

Online problems	Advice complexity	Paging	DiffServ	Conclusion

- problem complexity \approx amount of relevant information
- possible applications semi-online setting

Open directions

- other problems
- bounded advice
- trade-off between approximation and information
- randomization

Fhank you for your atention

・ロト ・聞 ト ・ ヨ ト ・ ヨ ト

Online problems	Advice complexity	Paging	DiffServ	Conclusion

- problem complexity \approx amount of relevant information
- possible applications semi-online setting

Open directions

- other problems
- bounded advice
- trade-off between approximation and information
- randomization

Fhank you for your atention

・ロト ・聞 ト ・ ヨ ト ・ ヨ ト

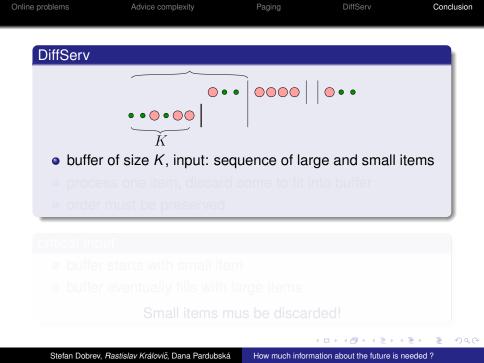
Online problems	Advice complexity	Paging	DiffServ	Conclusion

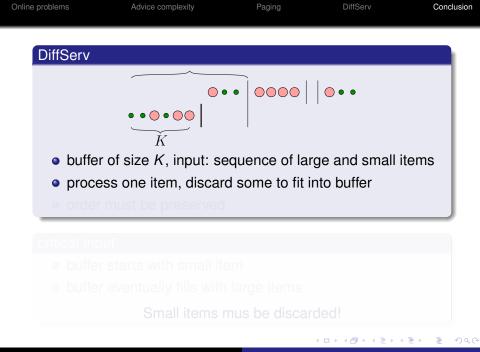
- problem complexity \approx amount of relevant information
- possible applications semi-online setting

Open directions

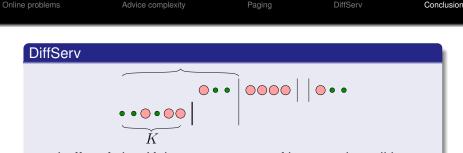
- other problems
- bounded advice
- trade-off between approximation and information
- randomization







Stefan Dobrev, Rastislav Královič, Dana Pardubská How much information about the future is needed ?

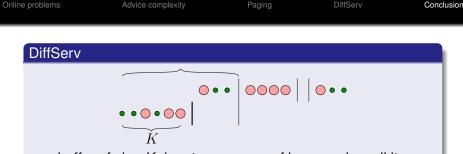


- buffer of size K, input: sequence of large and small items
- process one item, discard some to fit into buffer
- order must be preserved

- buffer starts with small item
- buffer eventually fills with large items

Small items mus be discarded!

イロト イポト イヨト イヨト



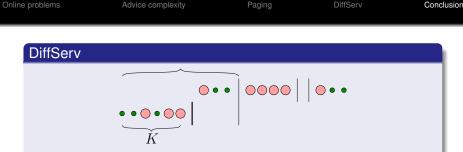
- buffer of size K, input: sequence of large and small items
- process one item, discard some to fit into buffer
- order must be preserved

- buffer starts with small item
- buffer eventually fills with large items

Stefan Dobrev, Rastislav Královič, Dana Pardubská How much information about the future is needed ?

・ロト ・四ト ・ヨト ・ヨト ・

ъ



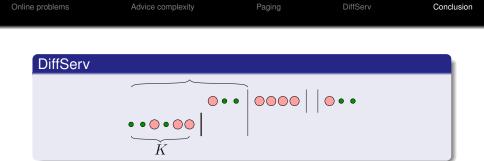
- buffer of size K, input: sequence of large and small items
- process one item, discard some to fit into buffer
- order must be preserved

- buffer starts with small item
- buffer eventually fills with large items

Small items mus be discarded!

・ロト ・回ト ・ヨト

ъ

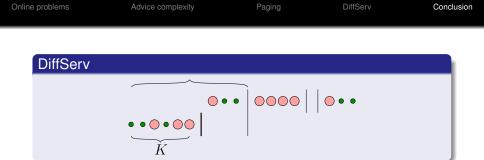


- buffer starts with small item
- buffer eventually fills with large items

upper bound

critical input at most ever K + 1 steps $\Rightarrow \frac{1}{K+1}$ bits per input

イロト イポト イヨト イヨト 一日



- buffer starts with small item
- buffer eventually fills with large items

upper bound

critical input at most ever K + 1 steps $\Rightarrow \frac{1}{K+1}$ bits per input

イロト 不得 とくほ とくほ とうほ