Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 140 (2005) 3–13
www.elsevier.com/locate/entcs

Variants of Online Chain Partition Problem of Posets[1]

Bartłomiej Bosek, Piotr Micek[2]

Computer Science Departament
Jagiellonian University
Kraków, Poland

Abstract

Online chain partitioning problem of posets is open for at least last 15 years. The best known online algorithm uses $\frac{5^w-1}{4}$ chains to cover width w posets. A variant of this problem considering only upgrowing posets, i.e. online posets in which every, new point is maximal at the moment it arrives, has been solved by Felsner [3]. He presented an algorithm using $\binom{w+1}{2}$ chains to cover posets of width w. Moreover, he proved that this is an optimal solution. We are interested in special class of posets, namely the interval posets. For this class we present an algorithm using $2w-1$ chains for width w and we prove that there is no better algorithm. In [3] Felsner also considers an adaptive chain covering problem for upgrowing posets. We are able to show a lowerbound for this problem to be $(2-\varepsilon)\cdot w$ for width w posets.

Keywords: partition, poset, game

1 Online Chain Partitioning Problem of Posets

For poset theoretic terminology we refer the reader to [9]. The basic problem we consider is an online chain partitioning problem. Kierstead formulated this problem in 1984. This problem can be viewed as a two-person game. We call the players Algorithm and Spoiler. Algorithm represents an online algorithm and Spoiler represents an adaptive adversary. The game is played in rounds. During round i the Spoiler introduces a new point x to the poset

[1] Extended abstract

[2] Email: bosek@ii.uj.edu.pl, micek@ii.uj.edu.pl

1571-0661 © 2005 Elsevier B.V. Open access under CC BY-NC-ND license.
doi:10.1016/j.entcs.2005.06.028

and describes comparabilities between x and points from previous rounds. The algorithm responds by assigning x to a chain. The most important thing in online problems is that previous moves(decisions) of Algorithm restrict his actual possibilities. The goal of Algorithm is to minimize the number of chains. Below we present an example of such a game.


first Spoiler's move

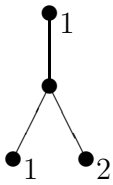
first Algorithm's move

second Spoiler's move

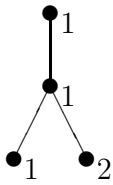
second Algorithm's move

third Spoiler's move

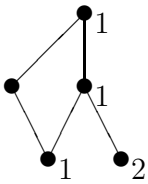
third Algorithm's move



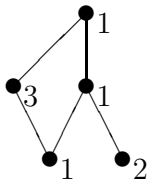
fourth Spoiler's move



fourth Algorithm's move

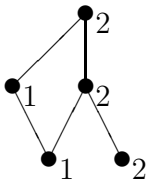


fifth Spoiler's move



fifth Algorithm's move

It is easy to see that presented poset actually be covered by chains $1, 2, 3$ can be covered by only two chains.



optimal cover

The partition presented above is optimal for this poset. We see a difference between online chain partitioning and "offline" version of this problem. In offline version all data about the poset are available to Algorithm on the spot, while in the online version he has to make his decisions without knowledge about incoming points. Thus, we are interested in the correspondence between the minimal number of chains needed to cover an online poset and the minimal number of chains needed to cover it "in natural way". From the next theorem we know that the second number is exactly the width of the poset

Definition 1.1 *We say that poset $\mathcal{P}$ has width$(\mathcal{P}) = w$ if there is an antichain in $\mathcal{P}$ of w points and there is no antichain in $\mathcal{P}$ of $w+1$ points.*

Theorem 1.2 (Dilworth) *If $\mathcal{P}$ is an poset and width$(\mathcal{P}) = w$, then there exists a partition $\mathcal{P} = \mathcal{C}_1 \cup \ldots \cup \mathcal{C}_w$, where $\mathcal{C}_i$ is a chain.*

On the other hand the value of the online chain partitioning problem for posets of width w is the largest integer $CP(w)$ so that there is a strategy of presenting points that forces any algorithm to use at least $CP(w)$ chains. Note that, we may as well define $CP(w)$ as the least integer so that there is an algorithm that never uses more chains. An argument of Szemeredi proves lowerbound and Kierstead[6] showed upperbound.

Theorem 1.3 (Kierstead[6]; Szemeredi)

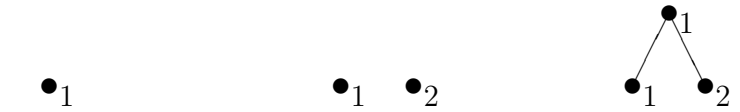
$$\binom{w+1}{2} \leqslant CP(w) \leqslant \frac{5^w - 1}{4}.$$

This is already a complicated result and no progress has been made on the general problem for the last 15 years.

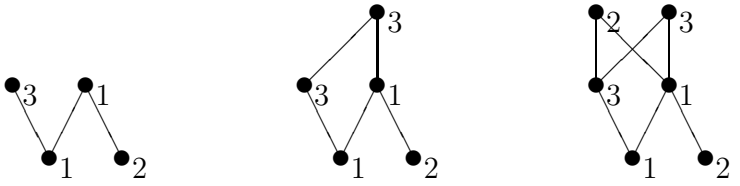
2 Upgrowing Version

Felsner [3] introduced a variant of the chain partitioning problem. In this problem we consider the same game as the previous one with one additional restriction. Every, new point x (presented by the Spoiler) has to be a maximal

point in a current poset of the game. Below we present an example for this variant.



three first moves



next three moves

Online posets with this property are called upgrowing posets. The value of the an online partitioning problem of upgrowing posets for posets width w defined similarly to previous one and it is denoted by $CPU(w)$. The upgrowing case is fully solved by the following theorem.

Theorem 2.1 (Felsner [3])

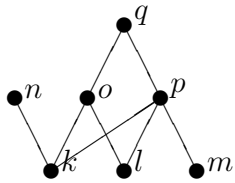
$$CPU(w) \;=\; \binom{w+1}{2}.$$

3 Interval Posets

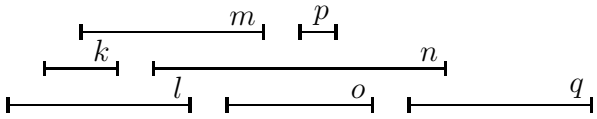
Online chain partitioning problem can be considered for special classes of posets. It turns out that the class of interval posets is an interesting class for this problem.

Definition 3.1 An poset $\mathcal{P}$ is called an interval poset if there is a function I assigning to each point $x \in \mathcal{P}$ a nondegenerate, closed interval $I(x) = [l_x, r_x]$ of the real line $\mathbb{R}$ so that $x < y$ in $\mathcal{P}$ if and only if $r_x < l_y$ in $\mathbb{R}$. The function I is called an interval representation of the poset $\mathcal{P}$.

The poset $\mathcal{Q}$



is an interval poset cause it has the following interval representation.

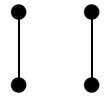


interval representation of $\mathcal{Q}$

Checking whether a partial poset is an interval poset directly from the definition may be complicated. Fortunately there are few equivalent conditions to the existence of an interval representation. To state one we need following definition.

Definition 3.2 An poset $\mathcal{P}$ is $2+2$-free if it has no elements $a, b, c, d \in \mathcal{P}$ with: $a < b$, $c < d$, $a \parallel d$, and $c \parallel b$, where $x \parallel y$ means that neither $x \leqslant y$ nor $y \leqslant x$.

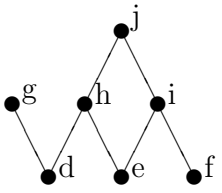
Loosely speaking in $2+2$-free poset there is no induced subposet which looks like:



poset $2+2$

Theorem 3.3 (Fishburn [4]) *A poset $\mathcal{P}$ is an interval poset if and only if the poset $\mathcal{P}$ is $2+2$-free.*

For example the poset $\mathcal{R}$



is not an interval poset, as the points d, f, g, i form a subposet of $\mathcal{R}$ isomorphic to 2+2.

At this moment mention some variants of online chain partition problem of interval posets. In the first variant Spoiler presents poset with representation, where new intervals are not necessary maximal at the moment they arrive. It is named online chain partition problem of interval posets with representation and the value of this game (problem) is denoted by $CPIR(w)$ for width w posets. M. Chrobak with M. Ślusarek prove lowerbound in [2]. Kierstead and Trotter showed upperbound.

Theorem 3.4 (Chrobak, Ślusarek [2]; Kierstead, Trotter [7])

$$CPIR(w) \;=\; 3w - 2.$$

This problem for first-fit algorithm was touched on for years. Linearity of an upperbound has been showed by H. Kierstead in [5] in 1988. It is known that the value of the best result is between $4.45w$ and $25.72w$ for width w posets. H. Kierstead with J. Qin in [8] showed lowerbound and M. Chrobak with M. Ślusarek in [2] showed upperbound.

In the second version as before, as now Spoiler presents poset with representation but new intervals have to be maximal at the moment they arrive. It is upgrowing version of the previous game. The value of the game is denoted by $CPUIR(w)$. P. Broniek showed strategy for Algorithm, which is as good as off-line solution.

Theorem 3.5 (Broniek [1])

$$CPUIR(w) \;=\; w.$$

Consider a version of the game where Spoiler presents points without representation. In other words, Spoiler has to present 2+2-free poset. Algorithm knows only relations between points. He has to assign a new point to a chain. This is an online chain partitioning problem of interval posets. The value of this game for width w posets is denoted by $CPI(w)$. Kierstead and Trotter proved

Theorem 3.6 (Kierstead, Trotter [7])

$$CPI(w) \;=\; 3w - 2.$$

Of course, there is an upgrowing version of this problem. Let $CPUI(w)$ be the value of this game. One of our result is:

Theorem 3.7

$$CPUI(w) \;=\; 2w - 1.$$

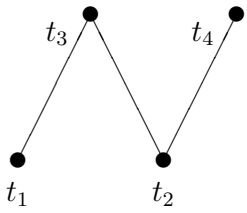
As we can see, considering on input only poset from a special class one may be able to use less chains to cover a poset than in general problem.

4 Adaptive Chain Partitioning Problem of Upgrowing Posets

We are looking for algorithms which schedule online tasks in a multiprocessor environment. Some tasks need as an input the outputs from previous tasks. Thus, the tasks cannot be performed in any poset, instead the poset has to follow the require data flow. Below we present an example:

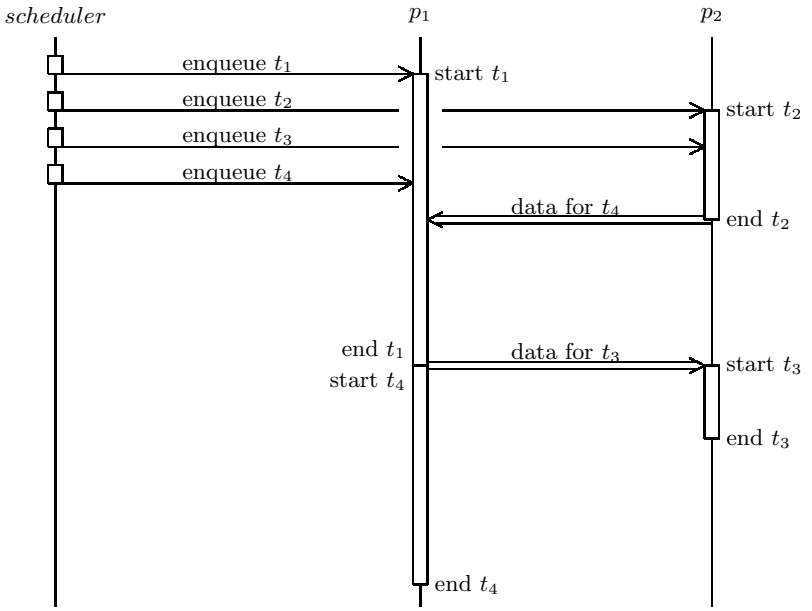
task	needs source of data from
t_1	
t_2	
t_3	$t_1,\ \ t_2$
t_4	t_1

Define $<$ as the relation between tasks so that $t_i < t_j$ if and only if t_j needs data from output of task t_i. It is easy to see that $<$ is a partail poset on the set of tasks. The poset described by to the example is presented below.



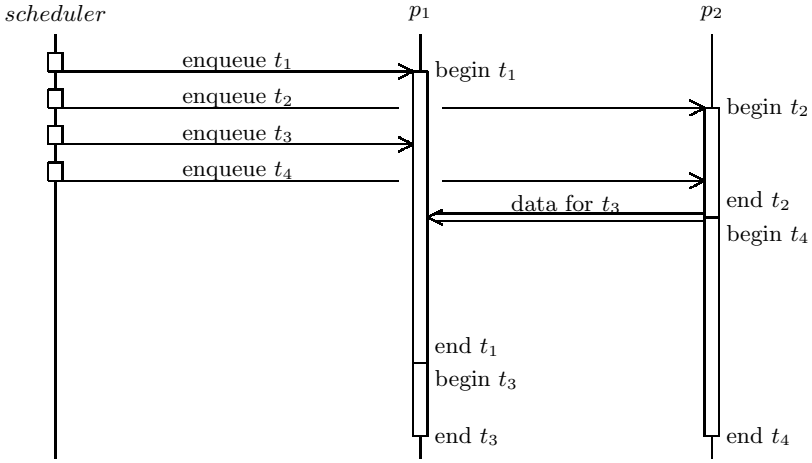
dependencies between t_1, t_2, t_3, t_4

At the moment a new task appears, scheduler must immediately assign a processor which will execute this task. It is unknown how long the execution will take and when the next task appears. Thus for each processor set of tasks scheduled to it, must form a chain. In other way we would lost optimality of execution's time. Indeed in our example with tasks t_1, t_2, t_3, t_4 one of the possibilities for the scheduler presented below.



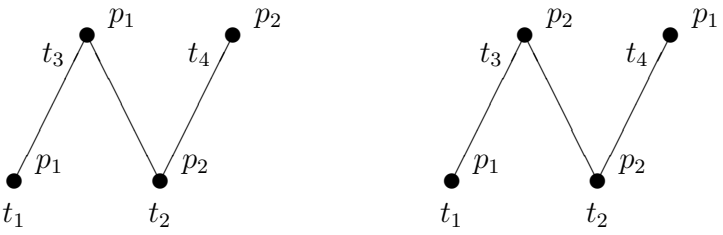
Solution 1: Data flow diagram - not optimal scheduler

The processor p_2 is inactive most of the time. Scheduler could have worked better in this case.



Solution 2: Data flow diagram - optimal scheduler

Obviously the task t_4 has to wait for the data from t_2. However, the first of the solutions, forces t_4 to unnecessarily wait for the processor p_1 to finish task t_1. The second solution avoids this unnecessary delay. Thus, if jobs scheduled to a processor form a chain we do not waste time, reaching an optimal solution. Using poset diagram, we may present the solutions 1 and 2 as follows.



Makespan for t_1, t_2, t_3, t_4 for correct and uncorrect solutions

It is easy to translate the scheduling problem to online chain partitioning problem of upgrowing posets. Hence, we may use Felsner's algorithm using $\binom{w+1}{2}$ chains for width w posets. But is this the best way to schedule tasks online? Till now it is concealed that one task may be scheduled to more than one processor. Each of the processors assigned to a particular task may perform this task independently. Scheduler may now remove a task from a processor if it is also executed/scheduled to another processor(s). Scheduler must choose online a group of processors for a new task and then afterwards he can only remove some but not all of them.

Now, define an online chain partitioning problem corresponding to this scheduling problem. Again, Spoiler presents an upgrowing poset. Algorithm may cover new point x by some non-empty set of chains and than during covering next points he may remove some chains from x. But for each point x there must be at least one chain covering it at any moment of the game. This problem was stated by Felsner in [3] and it is called an adaptive chain covering problem of upgrowing posets. As for previous games (problems) we define the value of the game for posets of width w and we define it by $ACPU(w)$ The natural upperbound for $ACPU(w)$ can be obtained immediately from Theorem 2.1:

$$ACPU(w) \leqslant CPU(w) = \binom{w+1}{2}.$$

Now the reader may ask whether the above inequality is strong. Consider two games. First one is a standard chain partitioning of upgrowing posets. The second one is an adaptive version of it.

STANDART VERSION

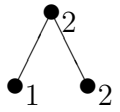
$\bullet_1$

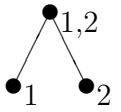
first moves

ADAPTIVE VERSION

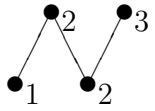
$\bullet_1$

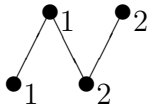
second moves





third moves - adaptivity





fourth moves - defence from thew strike

By Theorem 2.1 we know that $CPU(2) = 3$ but it turns out that in fact

$$ACPU(2) \;=\; 2.$$

Unfortunately the hipothesis that $ACPU(w) = w$ is false. The best known lowerbound for $ACPU(w)$ is presented below

Theorem 4.1

$$\limsup_{w\to\infty} \frac{ACPU(w)}{w} \;\geqslant\; 2.$$

Less formally, for each $\varepsilon > 0$ and sufficiently large w we have

$$(2-\varepsilon)\cdot w \;\leqslant\; ACPU(w).$$

As we can see there is a quite big gap between lowerbound $(2-\varepsilon)\cdot w$ and upperbound $\binom{w+1}{2}$ for $ACPU(w)$. This problem is still open and looks really challenging.

We want to thank our supervisor - Paweł Idziak for motivation and hours spent with us.

References

[1] P. Broniek, *On-line chain partitioning as a model for real-time scheduling*, this volume.

[2] M. Chrobak, M. Ślusarek *On some Packing Problem Related to Dynamic Storage Allocation* Theoretical Informatics and Applications 22:487-499, 1988.

[3] S. Felsner *On-line chain partitions of orders*, Theoretical Computer Science 175:283-292, 1997.

[4] P. C. Fishburn *Intrasitive indifference with unequal indifference intervals*, J. Math. Psych. 7:144-149, 1970.

[5] H. A. Kierstead *The linearity of first-fit coloring of interval graphs*, SIAM J. Discrete Math., 1:526-530, 1988.

[6] H. A. Kierstead, G. F. McNulty and W. T. Trotter. *A theory of recursive dimension for ordered sets*, Order, 1:67-82, 1984.

[7] H. A. Kierstead, W. T. Trotter. *An extremal problem in recursive combinatorics*, Proceedings of the Twelfth Southeastern Conference on Combinatorics, Graph Theory and Computing, Vol. II (Baton Rouge, La., 1981). Congr. Numer. 33 (1981), 143–153.

[8] H. A. Kierstead, J. Qin *Coloring interval graphs with First-Fit*, Discrete Math., 144:47-57, 1995.

[9] W. T. Trotter *Combinatorics and Partially Ordered Sets: Dimension Theory.* John Hopkins Press, 1992.