

On the performance of on-line algorithms for partition problems*

ULRICH FAIGLE,¹ WALTER KERN¹ and GYÖRGY TURÁN^{2,3}

¹*Faculty of Applied Mathematics, University of Twente NL-7500 AE Enschede,
The Netherlands*

²*Department of Mathematics, Statistics and Computer Science
University of Illinois at Chicago, Chicago, IL, 60680, USA*
and

³*Automata Theory Research Group of the Hungarian Academy of Sciences, Szeged, 6720, Hungary.*

Abstract

We consider the performance of the greedy algorithm and of on-line algorithms for partition problems in combinatorial optimization. After surveying known results we give bounds for matroid and graph partitioning, and discuss the power of non-adaptive adversaries for proving lower bounds.

1. Introduction

There are several combinatorial optimization problems where a set is to be partitioned into a minimal number of classes having certain properties. Examples of such problems are graph coloring and bin packing. A general heuristic to find an approximate solution is the greedy (or first-fit) method where the partition is constructed by processing the elements in some order and placing each element into the first class it fits into.

A partitioning algorithm is on-line if it considers the elements one after the other and puts each element into a class at the time when it is considered according to some rule, based on information about elements processed earlier (thus the greedy method is a special case). The main feature of an on-line algorithm is that the decision made about an element cannot be modified later on. An on-line algorithm in general does not have to be polynomial time computable or even computable.

There are several interesting results about the performance of on-line algorithms for various partition problems. After giving a general problem formulation in Section 2. we survey these results in Section 3.

* Supported by Hungarian Academy of Sciences. (OTKA Nr. 1135)

In Section 4. we consider the matroid partitioning problem and the special cases of graphic matroids and graphs. There are polynomial time algorithms solving this problem (Edmonds [6], see also Lawler [18]), but these algorithms are not on-line. We show that the performance ratio of the greedy algorithm on n element matroids is $\theta(\log n)$ and that the performance ratio of every on-line matroid partitioning algorithm is $\Omega(\log n/\log \log n)$. We also show that bounded performance is not possible even in the special case when we want to partition a graph into forests.

All known lower bound proofs for on-line algorithms are based on the construction of an adversary which plays against the algorithm by providing the new elements of the input so that the algorithm is forced to produce more classes than necessary. In many cases the adversary satisfies a condition called non-adaptiveness. In Section 5. we consider examples comparing the power of non-adaptive adversaries and general ones for lower bound proofs.

Section 6. contains some further remarks and open problems.

2. Partition problems, definitions

First we give a list of partition problems discussed later on. For definitions not given here see Bollobás [2], Lawler [18], Lovász [22], Welsh [30].

MATROID PARTITIONING: given a matroid $M=(E, \mathcal{F})$, partition the ground set E into a minimal number of independent subsets.

GRAPHIC MATROID PARTITIONING: the same as above for a graphic matroid M .

(As the complexity of the algorithms is not taken into consideration we may assume that the matroids are presented by listing their independent subsets.)

GRAPH PARTITIONING: given a graph $G=(V, E)$, partition E into a minimal number of forests.

GRAPH COLORING: given a graph $G=(V, E)$, partition V into a minimal number of independent subsets.

CHAIN DECOMPOSITION OF ORDERED SETS: given an ordered set $P=(V, <)$, partition V into a minimal number of chains.

GRAPH EDGE COLORING: given a graph $G=(V, E)$, partition E into a minimal number of matchings.

BIN PACKING: given $A=\{a_1, \dots, a_n\}$ ($0 < a_i \leq 1$), partition A into a minimal number of sets each having sum ≤ 1 .

GRAPH BIN PACKING: given a fixed "pattern" graph $G_0=(V_0, E_0)$ and a graph $G=(V, E)$, partition E into a minimal number of sets each being a subgraph of G_0 .

A common framework for considering these problems can be described using independence systems.

An independence system is a pair $I=(E, \mathcal{F})$, where E is the ground set and $\mathcal{F} \subseteq \mathcal{P}(E)$ is a set of subsets of E such that if $F \in \mathcal{F}$ and $F' \subseteq F$ then $F' \in \mathcal{F}$. An independence system is ordered if in addition there is a linear ordering $<$ on E . All ordered independence systems considered here are finite, we write $I_n=(E_n, \mathcal{F}_n)$, $E_n=\{e_1, \dots, e_n\}$, $e_1 < \dots < e_n$. An ordered independence system $I_k=(E_k, \mathcal{F}_k)$ is an initial segment of I_n (denoted by $I_k < I_n$) if $k \leq n$, $E_k=\{e_1, \dots, e_k\}$ and for every $F \subseteq E_k$ it holds that $F \in \mathcal{F}_k$ iff $F \in \mathcal{F}_n$.

An independent partition of $I=(E, \mathcal{F})$ is an ordered partition (F_1, \dots, F_l) of E such that $F_i \in \mathcal{F}$ ($1 \leq i \leq l$). Let $p(I) := \min \{l : \text{there is an independent partition } (F_1, \dots, F_l) \text{ of } E\}$.

Let \mathcal{S} be a class of finite independence systems. The PARTITION PROBLEM FOR \mathcal{S} is the following problem: given $I=(E, \mathcal{F}) \in \mathcal{S}$, find an independent partition of E into $p(I)$ sets.

Assume that furthermore \mathcal{S} consists of ordered independence systems and is closed under taking initial segments (i.e. $I \in \mathcal{S}$, $I' < I$ imply $I' \in \mathcal{S}$).

An on-line algorithm A for the partition problem for \mathcal{S} is a function defined on \mathcal{S} such that for every $I=(E, \mathcal{F}) \in \mathcal{S}$ $A(I)$ is an ordered independent partition of I and if $I'=(E', \mathcal{F}') < I$ then $A(I')=A(I)|_{E'}$ i.e. $A(I')$ is the restriction of $A(I)$ to E' , or equivalently, $A(I)$ is an extension of $A(I')$. Thus A provides an approximate solution to the partition problem for \mathcal{S} .

For the greedy algorithm A_{gr} , $A_{gr}(I_n)$ is obtained from $A_{gr}(I_{n-1})$ by placing e_n into the first subset in the ordered partition $A_{gr}(I_{n-1})$ which remains independent if e_n is added to it, and opening a new set for e_n if there is no such set.

For an on-line algorithm A let $|A(I)|$ be the number of subsets in the partition $A(I)$ and let (with some abuse of notation)

$$A(n) := \max \{|A(I)|/p(I) : I = (E, \mathcal{F}) \in \mathcal{S}, |E| = n\}$$

be the performance ratio function of A . A has bounded performance with bounding function $f: \mathbf{N} \rightarrow \mathbf{N}$ if for every $I \in \mathcal{S}$ it holds that $|A(I)| \leq f(p(I))$. (Thus if $A(n) \rightarrow \infty$ by considering inputs with $p(I)$ bounded by some constant then A does not have bounded performance.) The performance ratio of A is

$$r_A := \inf \{r \geq 1 : |A(I)| \leq rp(I) \text{ for every } I \in \mathcal{S}\}$$

and the asymptotic performance ratio of A is

$$r_A^\infty := \inf \{r \geq 1 : \exists c : |A(I)| \leq rp(I) + c \text{ for every } I \in \mathcal{S}\}$$

(thus $r_A, r_A^\infty \in \mathbf{R} \cup \{\infty\}$). Let

$$r_{\mathcal{S}} := \inf \{r_A : A \text{ is an on-line algorithm for the partition problem for } \mathcal{S}\},$$

$$r_{\mathcal{S}}^\infty := \inf \{r_A^\infty : A \text{ is an on-line algorithm for the partition problem for } \mathcal{S}\}.$$

For matroid partitioning (resp. graphic matroid partitioning) the class \mathcal{S} could be the class of all finite ordered matroids (resp. finite ordered graphic matroids) on a fixed countable set. For graph partitioning the class \mathcal{S} could consist of all finite (edge-)ordered subgraphs of a countable complete graph. For bin packing the class \mathcal{S} could consist of all finite ordered subsets of countably many copies of $(0, 1]$.

There is a difference between the first two and the last two examples. For matroid partitioning and graphic matroid partitioning we may assume that if $I_1 < I_2 < \dots < I_n$, $I'_1 < I'_2 < \dots < I'_n$ and $I_j \cong I'_j$ ($1 \leq j \leq n$) then the partitions determined by A are also isomorphic, in particular $|A(I_n)| = |A(I'_n)|$. This holds because \mathcal{S} is homogeneous, i.e. every isomorphism of two inputs I and I' can be extended to an automorphism of \mathcal{S} . With other words on-line algorithms for these problems can only use information about independence.

For the other two problems there is additional information provided by specify-

ing edges resp. numbers: 2 edges with or without a common endpoint are isomorphic as independence systems but none of their isomorphisms can be extended to an automorphism of \mathcal{I} ; resp. there is no automorphism of \mathcal{I} for bin packing mapping a copy of $1/2$ to a copy of $1/3$.

3. A survey of results about on-line algorithms

a) Graph coloring

Johnson [12] observed that $A_{\text{gr}}(n) = \Omega(n)$ even for bipartite graphs. Szegedy [25] showed that for every on-line graph coloring algorithm $A(n) = \Omega(n/(\log n)^2)$. Lovász, Saks and Trotter [23] gave an on-line algorithm with $A(n) = O(n/\log^* n) = o(n)$. For trees Bean [1] and Gyárfás and Lehel [11] noted that $A(n) = \Omega(\log n)$ for every on-line algorithm. Kierstead and Trotter [16] gave an on-line algorithm coloring interval graphs with $r_A^\infty = 3$ and showed that this is best possible. Kierstead [15] showed that for this problem $r_{A_{\text{gr}}}^\infty < \infty$. Gyárfás and Lehel [11] showed that $r_{A_{\text{gr}}}^\infty < \infty$ for several special classes of graphs such as split graphs, complements of bipartite graphs and complements of chordal graphs.

b) Chain decomposition of ordered sets

Kierstead [14] proved that there is an on-line algorithm for this problem which has bounded performance with bounding function $(5^n - 1)/4$. This appears to be the first result on on-line algorithms formulated in the language of recursion theoretic combinatorics. For the greedy algorithm $A_{\text{gr}}(n) = \Omega(n)$. Szemerédi [26] showed that for every on-line algorithm A and every w there are orders P with width w and $|A(P)| = \Omega(w^2)$ thus for every on-line algorithm A $r_A^\infty = \infty$. An order is an interval order if it is isomorphic to a set of intervals $\{J_1, \dots, J_n\}$ on a line with $J_i < J_j$ iff J_i is completely to the left of J_j . Kierstead and Trotter [16] gave an on-line algorithm for interval orders with $r_A^\infty = 3$ and showed that this is optimal. (We note that the difference between the chain decomposition problem and the graph coloring problem for incomparability graphs is that comparable pairs form an ordered resp. an unordered pair.) An order is series-parallel if it can be obtained from orders on one element by repeated application of series composition ("place order P_1 above P_2 ") and parallel composition ("let all elements of P_1 be incomparable to all elements of P_2 "). If the orders are restricted to be series-parallel then the greedy algorithm always gives an optimal solution [7].

c) Graph edge coloring

If Δ is the maximal degree of the graph $G = (V, E)$ then clearly $\cong \Delta$ colors are needed for an edge coloring of G (by Vizing's theorem [29] (see also Bollobás [2]) $\Delta + 1$ colors are always sufficient). It is easy to see that the greedy algorithm never uses more than $2\Delta - 1$ colors. On the other hand every on-line algorithm A uses $\cong 2\Delta - 1$ colors for some forest with maximal degree Δ (here the minimal number of colors needed is easily seen to be Δ). To see this, consider first a forest of $(\Delta - 1) \cdot \binom{2\Delta - 2}{\Delta - 1} + 1$ stars with $\Delta - 1$ edges. Then A either uses $\cong 2\Delta - 1$ colors or there will be Δ stars colored with the same set of $\Delta - 1$ colors. Add Δ new edges by connecting a new root to the root of these stars to get a forest with maximal degree Δ .

Every new edge must be colored with a color not occurring in the stars selected and thus $\cong 2\Delta - 1$ colors will be used.

d) Bin packing

Johnson, Demers, Ullman, Garey and Graham [13] showed that $r_{A_{gr}}^\infty = 1.7$. Yao [31] gave an on-line algorithm with $r_A^\infty = 5/3$. The on-line algorithm of Lee and Lee [19] has $r_A^\infty \cong 1.692$ and it also satisfies the additional requirement of having only a bounded number of active bins at any time. Brown [4] and Liang [20] showed that $r_A^\infty \cong 1.536$ for every on-line algorithm. This result is generalized by Galambos [8] to the case when items are from $(0, \alpha]$ ($\alpha < 1$). We note that there are polynomial time algorithms A_ε (which are not on-line) with $r_{A_\varepsilon} < 1 + \varepsilon$ for every $\varepsilon > 0$. (de la Vega and Lueker [28]). On-line algorithms for dual bin packing (where the aim is to fill as many bins as possible) are considered by Csirik and Totik [5]. For the graph bin packing problem it is shown in [27] that for complete bipartite graphs $G_0 = K_{k,l}$, $k \leq l$, $r_{A_{gr}} = \Theta(\max(k, l/k))$, thus for fixed l the greedy algorithm has the best performance guarantee when $k \sim \sqrt{l}$.

We note that there are results about on-line algorithms for problems of a different nature than the ones discussed here (see Borodin, Linial and Saks [3], Manasse, McGeoch and Sleator [24] and the further references in these papers).

4. Matroid partitioning

First we consider the performance of the greedy algorithm. The upper bound holds for matroids in general, the lower bound already holds in the special case of graphs.

- Theorem 1.** a) For the matroid partitioning problem $A_{gr}(n) \cong \ln(n)$.
 b) For the graph partitioning problem $A_{gr}(n) \cong \lfloor \log n \rfloor / 2$.

Proof. a) Let $I_n = (E_n, \mathcal{F}_n)$ be a matroid and (F_1, \dots, F_l) be the partition formed by the greedy algorithm. Then F_i is a maximal independent set in $E_n \setminus (F_1 \cup \dots \cup F_{i-1})$. As I_n restricted to $E_n \setminus (F_1 \cup \dots \cup F_{i-1})$ is again a matroid, F_i is also a maximum independent set in $E_n \setminus (F_1 \cup \dots \cup F_{i-1})$. Thus (F_1, \dots, F_l) is a greedy solution of the set covering problem for I_n . The performance ratio of the greedy algorithm for set covering is $\cong \ln(n)$ (Johnson [12], Lovász [21]).

b) For $k \geq 1$ let $G_k := (V_k, E_k)$, where

$$V_k = \{v_0, \dots, v_{2^k-1}\}, \quad E_k = \bigcup_{i=0}^{k-1} P_i,$$

$$P_i = \{(v_{j2^i}, v_{(j+1)2^i}) : j = 0, \dots, 2^{k-1-i} - 1\}.$$

For later use let v_0 be the initial vertex of G_k and v_{2^k-1} be the terminal vertex of G_k . Order the edges in G_k in such a way that edges in P_i precede edges in P_{i+1} ($0 \leq i \leq k-2$). Then the greedy algorithm gives a different color to each P_i (we refer to this partition of E_k as the greedy partition), hence for this ordering $|A_{gr}(G_k)| = k$. Note that $|E_k| = 2^k - 1$. On the other hand coloring the edges of P_i alternately red and blue (for every i) gives a partition of E_k into 2 trees and so $p(G_k) = 2$. \square

Theorem 1. can be generalized to the case when \mathcal{I} consists of independence systems that are the intersections of k matroids (thus for every $I=(E, \mathcal{F}) \in \mathcal{I}$ there are k matroids $I^i=(E, \mathcal{F}^i)$ ($1 \leq i \leq k$) such that for every $F \subseteq E$, $F \in \mathcal{F}$ iff $F \in \mathcal{F}^i$ for every $i=1, \dots, k$).

Corollary 2. Assume that for every $I \in \mathcal{I}$, I is the intersection of k matroids. Then for the partitioning problem for \mathcal{I} it holds that $A_{gr}(n) \leq k \cdot \ln(n)$.

Proof. Korte and Hausmann [17] showed that if $I=(E, \mathcal{F})$ is the intersection of k matroids, F is a maximal independent set in \mathcal{F} and F' is a maximum independent set in \mathcal{F} then $|F| \geq (1/k) \cdot |F'|$. Thus the partition given by the greedy algorithm is a "1/k-greedy" solution to the set covering problem on I in the sense that we always choose a set which has size $\geq 1/k$ times the size of a largest set in the system. The proof of Johnson [12] and Lovász [21] can be applied to this case to show that the number of sets used in the covering is $\leq k \cdot \ln(n)$ times the optimal. \square

Now we turn to the discussion of on-line algorithms.

Theorem 3. For every on-line matroid partitioning algorithm A $A(n) = \Omega(\log n / \log \log n)$.

Proof. For G_i constructed in the proof of Theorem 1. let $s \cdot G_i$ be the graph obtained by taking a sequence of s copies of G_i and identifying the terminal vertex of each copy (except the last one) with the initial vertex of the next one.

For a graph G let $M(G)$ be the cycle matroid of G .

Then $M(sG_i)$ is the direct sum of s copies of $M(G_i)$. (The direct sum of matroids on disjoint ground sets is obtained by taking the union of the ground sets as the new ground set and letting a subset be independent if its intersection with each ground set is independent.) If M is a matroid isomorphic to $M(sG_i)$ then it has a unique decomposition into s matroids isomorphic to $M(G_i)$, called the components of M . An ordered partition of M into independent subsets is called the greedy partition if on each component it corresponds to the greedy partition of G_i .

The graph $2G_i$ is a subgraph of G_{i+1} and therefore a matroid $M \cong M(G_i) \oplus M(G_i)$ (where \oplus denotes the direct sum) can be extended to a matroid isomorphic to $M(G_{i+1})$ by adding one more element to it.

Now let $g(1) := 1$, $g(k) := (k-1)(2g(k-1)-1)+1$ for $k > 1$ and $f(k) := \sum_{i \leq k} g(i)$ for $k \geq 1$.

We show that the algorithm A uses $\geq k$ colors to partition some 2-partitionable matroid on $f(k)$ elements.

Using an adversary strategy we prove that giving $g(k) + \dots + g(k-i)$ elements ($0 \leq i \leq k-2$) to A it can be forced either to use $\geq k$ colors or to form the greedy partition on a submatroid isomorphic to $M(2g(k-i-1)G_{i+1})$.

For $i=0$, giving $g(k)$ independent elements to A it either uses $\geq k$ colors or it assigns the same color to $2g(k-1)$ elements and $M(G_i)$ consists of a single element.

For the induction step assume that after adding $g(k) + \dots + g(k-i+1)$ elements to A it formed the greedy partition on a submatroid $M_i \cong M(2g(k-i)G_i)$. Pair the components of M_i and add $g(k-i)$ elements (one to each pair) to extend each pair to a matroid isomorphic to $M(G_{i+1})$. As A cannot use any of the i colors used for M_i it either uses $\geq k-i$ colors different from these or it assigns the same

color to $2g(k-i-1)$ new elements. The union of these components is $M_{i+1} \cong M(2g(k-i-1)G_{i+1})$ and A formed the greedy partition on M_{i+1} .

For $i=k-2$ we get $M_{k-1} \cong M(2G_{k-1})$ such that A formed the greedy partition on M_{k-1} . Adding a new element to obtain $M_k \cong M(G_k)$ forces A to use the k^{th} color.

As the components of the matroid M formed by all elements given to A are isomorphic to $M(G_i)$ for some i , M is 2-partitionable.

Finally the bound follows from noting that $g(k) \leq 2kg(k-1)$, thus $g(k) \leq 2^k \cdot k!$. Hence $f(k) \leq 2^k \cdot k \cdot k!$ and so $k = \Omega(\log n / \log \log n)$. \square

Corollary 4. For every on-line algorithm A partitioning graphic matroids $A(n) = \Omega(\log n / \log \log n)$.

Proof. All matroids constructed in the previous proof are graphic. \square

We remark that the proof of Theorem 3. does not work for graphs. This is related to the remarks made following the definitions in Section 2. For graphs the adversary is in a more difficult situation as e.g. 2 independent elements in the first phase of the construction can be completed to a triangle by adding a new element if we are dealing with general (or graphic) matroids but in graphs this can only be done if the 2 edges have a common endpoint.

Let $g(1) := 1$, $g(k) := (2k)^{(k-1)(2g(k-1)-1)+1} - 1$ for $k > 1$ and $f(k) := \sum_{i \leq k} g(i)$ for $k \geq 1$.

Theorem 5. Every on-line graph partitioning algorithm A forms at least k classes for some 2-partitionable graph having $f(k)$ edges.

Proof. We describe an adversary strategy by induction on k , for $k=1$ the statement is obvious. First we prove a lemma.

Lemma 6. For every l ($2 \leq l \leq k$), by building a forest on $g(l)+1$ vertices A can be forced either to use at least l colors or to form a monochromatic path P of length $2g(l-1)$.

Proof. A forest is rooted if each of its components has a distinguished vertex called the root. An l -edge colored rooted forest with j roots is an (i, j) -forest if there are numbers t_1, \dots, t_l with $t_1 + \dots + t_l = i$ such that for every root v and every r ($1 \leq r \leq l$) v is the endpoint of a monochromatic path of color r and length t_r .

We show that for every $i=0, \dots, (l-1)(2g(l-1)-1)+1$ by building a forest A can be forced either to use $\geq l$ colors or to form an $(i, (g(l)+1)/(2l)^i)$ -forest.

For $i=0$ the empty graph on $g(l)+1$ vertices is a $(0, g(l)+1)$ -forest. Assume we constructed an $(i-1, (g(l)+1)/(2l)^{i-1})$ -forest. Add $(g(l)+1)/(2(2l)^{i-1})$ new edges forming a matching of the roots. Then A either uses $\geq l$ colors to color these edges or $\cong (g(l)+1)/(2l)^i$ new edges get the same color. In this case select an endpoint of each of these edges and let them be the new roots. Deleting the components without a selected root we get an $(i, (g(l)+1)/(2l)^i)$ -forest and the whole graph built is a forest.

For $i=(l-1)(2g(l-1)-1)+1$ we get an $(i, 1)$ -forest, i.e. a tree with $t_1 + \dots + t_l = (l-1)(2g(l-1)-1)+1$. Thus for some r ($1 \leq r \leq l$) it holds that $t_r \geq 2g(l-1)$. The path P required can be chosen to be the corresponding path of color r . \square

Now we describe the adversary strategy S_k .

1) Force A either to use $\cong k$ colors or to form a monochromatic path P of length $2g(k-1)$ by building a forest on a set V_k of $g(k)+1$ vertices. (This can be done by Lemma 6.)

2) Apply S_{k-1} to the set V_{k-1} consisting of every second vertex of P (thus $|V_{k-1}|=g(k-1)+1$).

Note that after completing phase 1) V_{k-1} is an independent set of vertices and in later stages the color of the path P cannot be used as otherwise a monochromatic cycle is created. Thus by induction S_k indeed forces A to use $\cong k$ colors and the construction implies that the graph G built by the adversary has $\cong f(k)$ edges.

Finally we claim that G is 2-partitionable. This follows by induction. Assume that the graph G' built on V_{k-1} is 2-partitionable and let (F_1, F_2) be a partition of its edges into 2 forests. Then adding the edges of P to F_1 and F_2 alternately and adding the remaining edges of G arbitrarily we get a 2-partition of G . \square

By definition, Theorem 5. implies the following.

Corollary 7. For every on-line graph partitioning algorithm A $A(n) \rightarrow \infty$ and A does not have bounded performance. \square

5. Non-adaptive adversaries

Several lower bounds for on-line algorithms are based on the existence of instances I such that for every independent partition of I there is an initial segment of I for which the restriction of the partition is far from being optimal. This shows that no on-line algorithm can have good performance on every initial segment of I .

Thus the adversary providing I is non-adaptive in the sense that for every algorithm A it provides a counterexample which depends on A in a very restricted way only through the choice of the initial segment of I . With other words the only liberty the adversary has is to decide when to stop giving new elements.

All known lower bounds for bin packing are non-adaptive. On the other hand the lower bounds for graph coloring and chain decomposition (e.g. [25], [14], [16]), and the lower bounds of the preceding section are adaptive, i.e. when the adversary determines the next extension of the current instance it takes into consideration the previous decisions made by the algorithm.

For $I_n=(E_n, \mathcal{F}_n)$ let $I_1 < \dots < I_n$ be the initial segments of I_n , $P_n=(F_1, \dots, F_d)$ be an independent partition of E_n and $P_k=P_n|E_k$ ($1 \leq k \leq n$) be the restriction of P_n to E_k . With these notations let

$$s_{\mathcal{F}}^{\infty} := \inf \{r: \exists c \forall I_n \in \mathcal{I} \exists P_n \forall P_k: |P_k| \leq rp(I_k) + c\}$$

($s_{\mathcal{F}}$ could be defined analogously). By the argument above $s_{\mathcal{F}}^{\infty} \leq r_{\mathcal{F}}^{\infty}$. We consider the question of how good a lower bound is $s_{\mathcal{F}}^{\infty}$ to $r_{\mathcal{F}}^{\infty}$.

For graph coloring restricted to forests clearly $s_{\mathcal{F}}^{\infty}=1$ and as mentioned in Section 3. $r_{\mathcal{F}}^{\infty}=\infty$ (as $A(n)=\Omega(\log n)$ for every on-line algorithm). We mention another example where both $s_{\mathcal{F}}^{\infty}$ and $r_{\mathcal{F}}^{\infty}$ are finite but different.

As it is mentioned in Section 3., Kierstead and Trotter [16] showed that $r_{\mathcal{F}}^{\infty} = 3$ for the chain decomposition problem restricted to interval orders.

Proposition 8. For the chain decomposition problem restricted to interval orders $s_{\mathcal{F}}^{\infty} \leq 2$.

Proof. The bound follows directly from the proof of Kierstead and Trotter [16]. Let P be an interval order of width w on the ground set $V = \{v_1, \dots, v_n\}$. Then V is partitioned into w sets L_1, \dots, L_w by considering the elements v_1, \dots, v_n one after the other and putting each element into the first set so that the conditions $\text{width}(P_n | L_1 \cup \dots \cup L_i) = i$ remain satisfied for every $i \leq w$ such that $L_i \neq \emptyset$. It is shown in [16] that then $\text{width}(L_i) \leq 2$ for every $i \leq w$. The proposition follows by considering a chain decomposition of P which consists of the chain L_1 and ≤ 2 chains covering L_i for $2 \leq i \leq w$. \square

Now we give an example where $s_{\mathcal{F}}^{\infty} = r_{\mathcal{F}}^{\infty}$.

Let RESTRICTED BIN PACKING be the bin packing problem restricted to items with sizes $(1/2) - \varepsilon$ and $(1/2) + \varepsilon$ (for some fixed $\varepsilon < 1/6$). We denote $(1/2) - \varepsilon$ by a and $(1/2) + \varepsilon$ by b .

Theorem 9. For the restricted bin packing problem $s_{\mathcal{F}}^{\infty} = r_{\mathcal{F}}^{\infty} = 4/3$.

Proof. The lower bound $s_{\mathcal{F}}^{\infty} \geq 4/3$ is noted e.g. in Liang [20]. Consider $I' < I$ where I contains n a -items followed by n b -items and I' is the first half of I . If an algorithm A fills k bins with 2 a -items each after processing I' then

$$|A(I')|/p(I') = 2 - 2(k/n), \quad |A(I)|/p(I) \geq 1 + (k/n)$$

which implies the bound for $s_{\mathcal{F}}^{\infty}$.

To prove the upper bound we describe an on-line algorithm with $r_{\mathcal{F}}^{\infty} = 4/3$.

We distinguish 4 types of bins: a -bins, b -bins, aa -bins and ab -bins, corresponding to the items contained in the bin. The algorithm will also pair some bins, the possible bin-pair types will be (aa, a) , (aa, b) , and (aa, ab) . If a bin is not paired with any other bin it is called unpaired.

A new element is processed according to the following rules:

- a) for a new element a^* :
 - if there is a b -bin B then put a^* into B
 - else if there is an unpaired a -bin B then put a^* into B
 - else if there is an unpaired aa -bin B then put a^* into a new bin B' and pair B and B'
 - else open a new bin B for a^* ;
- b) for a new element b^* :
 - if there is an a -bin B then put b^* into B
 - else if there is an unpaired aa -bin B then put b^* into a new bin B' and pair B and B'
 - else open a new bin B for b^* .

If there are several bins satisfying a condition then the choice is arbitrary, for definiteness let us always choose the first one.

It is easy to see that all possible bin-pair types that may be formed by the algorithm are indeed (aa, a) , (aa, b) and (aa, ab) .

Let us assume that after processing a list I the algorithm created c_1 unpaired a -bins, c_2 unpaired b -bins, c_3 unpaired aa -bins, c_4 unpaired ab -bins, c_5 (aa, a) bin-pairs, c_6 (aa, b) bin-pairs and c_7 (aa, ab) bin-pairs.

By definition

$$|A(I)| = c_1 + c_2 + c_3 + c_4 + 2c_5 + 2c_6 + 2c_7, \quad (1)$$

as the number of b -items is a lower bound to $p(I)$

$$p(I) \cong c_2 + c_4 + c_6 + c_7, \quad (2)$$

and as the half of the number of items is a lower bound to $p(I)$

$$p(I) \cong (1/2)c_1 + (1/2)c_2 + c_3 + c_4 + (3/2)c_5 + (3/2)c_6 + 2c_7. \quad (3)$$

Subtracting (2) resp. (3) from (1) we get

$$|A(I)| - p(I) \cong c_1 + c_3 + 2c_5 + c_6 + c_7, \quad (4)$$

$$|A(I)| - p(I) \cong (1/2)c_1 + (1/2)c_2 + (1/2)c_5 + (1/2)c_6. \quad (5)$$

We note that there cannot be both an a -bin and a b -bin in the packing as in this case the item arriving later would not be put into a separate bin.

Lemma 10. $c_1 + c_3 + c_6 \cong 1$.

Proof. We consider 6 different cases.

1) There cannot be 2 unpaired a -bins as otherwise the a -item arriving later would not have to be put in a separate bin.

2) There cannot be an unpaired a -bin B and an unpaired aa -bin B' . Indeed, if the a -item in B comes last, then B could be paired with B' , if one of the a -items in B' comes last then before the arrival of this element we get a contradiction to 1).

3) There cannot be 2 unpaired aa -bins as otherwise before the arrival of the last item we get a contradiction to 2).

4) There cannot be an unpaired a -bin and an (aa, b) bin-pair by the remark preceding the lemma.

5) There cannot be an unpaired aa -bin and an (aa, b) bin-pair. Again by the remark preceding the lemma the item coming last must be the b -item. But then before the arrival of this item we get a contradiction to 3).

6) There cannot be 2 (aa, b) bin-pairs. Again, the last item arriving must be a b -item. But then before the arrival of this item we get a contradiction to 5). \square

In the proof of the theorem we distinguish 2 cases.

Case 1. $c_2 = 0$.

Then using Lemma 10., (5) and $c_5 \cong (2/3)p(I)$ following from (3) we get

$$|A(I)| - p(I) \cong (1/2)c_5 + (1/2) \cong (1/3)p(I) + (1/2)$$

hence

$$|A(I)| \cong (4/3)p(I) + (1/2).$$

Case 2. $c_2 > 0$.

From the remark preceding Lemma 10. in this case $c_5 = 0$ and so we get from (4) and (5) using Lemma 10.

$$|A(I)| - p(I) \leq 1 + c_7 \quad (6)$$

$$|A(I)| - p(I) \leq (1/2) + (1/2)c_2. \quad (7)$$

Adding (7) twice and (6) and using $c_2 + c_7 \leq p(I)$ (cf. (2))

$$3(|A(I)| - p(I)) \leq 2 + c_2 + c_7 \leq 2 + p(I)$$

and so

$$|A(I)| \leq (4/3)p(I) + (2/3). \quad \square$$

6. Some remarks and problems

1. (Greedy algorithm vs. on-line algorithms.)

The chain decomposition problem for series-parallel orders is an example where the greedy algorithm gives an optimal solution. For the edge coloring problem $r_{A_{gr}}^\infty = 2$ and no on-line algorithm can have better performance. Thus for these problems on-line algorithms cannot perform better than the greedy algorithm.

On-line algorithms give a large improvement for the general chain decomposition problem (where $A_{gr}(n) = \Omega(n)$ and there is an on-line algorithm with bounded performance), for the graph coloring problem (where $A_{gr}(n) = \Omega(n)$ and there is an on-line algorithm with $A(n) = o(n)$) and for the bin packing problem (where $r_{A_{gr}}^\infty = 1.7$ and there is an on-line algorithm with $r_A^\infty = 5/3$).

There appears to be no example known where the greedy algorithm is not optimal but there is an on-line algorithm giving an optimal solution. Also for none of the examples considered does it hold that $r_{A_{gr}}^\infty = \infty$ but there is an on-line algorithm A with $r_A^\infty < \infty$.

2. (Bounds for particular problems.)

It would be interesting to improve the bounds for the performance of on-line algorithms for matroid and graph partitioning, in particular to decide if on-line algorithms can perform better than the greedy algorithm for partitioning graphs.

Concerning adversaries it appears to be not known if adaptive adversaries can lead to stronger lower bounds for the bin packing problem. Another question is the following: is $s_{gr}^\infty = \infty$ for the graph coloring problem? (Coloring optimally with i new colors those initial segments for which the chromatic number is i gives a coloring which uses $\leq i(i+1)/2$ colors for every initial segment of chromatic number i .)

A related partition problem which does not fit into the class of problems discussed here but which would be interesting to study in the context of on-line algorithms is the m -machine scheduling problem: given n tasks with execution times t_1, \dots, t_n find a schedule for m machines to minimize finishing time (thus here the number of the classes is fixed and we want to minimize the maximal weight). The greedy algorithm has performance ratio $2 - (1/m)$ (Graham [10]). No on-line algorithm appears to be known which improves this for any m . The lists (1, 1, 2) and (1, 1, 1,

3, 3, 3, 6) show that no improvement is possible for $m=2$ and $m=3$. The list (1 m times, $1 + \sqrt{2}$ m times, $2(1 + \sqrt{2})$ once) shows that $1 + (1/\sqrt{2})$ is a lower bound for the performance ratio of on-line algorithms for every $m \geq 4$.

Acknowledgement. We thank Collette Coullard, János Csirik, Gábor Galambos and László Lovász for their valuable remarks.

References

- [1] D. BEAN: Effective coloration, *J. Symb. Logic* 41 (1976), 469—480.
- [2] B. BOLLOBÁS: *Extremal Graph Theory*, Academic Press, London, 1976.
- [3] A. BORODIN, N. LENIAL, M. SAKS: An optimal on-line algorithm for for metrical task systems, 19. STOC (1987), 373—382.
- [4] D. J. BROWN: A lower bound for on-line one-dimensional bin packing algorithms, Tech. Rep. No. R-864, Coord. Sci. Lab., Univ. of Illinois, Urbana, IL, 1979.
- [5] J. CSIRIK, V. TOTIK: On-line algorithms for a dual version of bin packing, *Discr. Appl. Math.* 21 (1988), 163—167.
- [6] J. EDMONDS: Minimum partition of a matroid into independent subsets, *J. Res. Nat. Bur. St.* 69B (1965), 67—72.
- [7] U. FAIGLE, GY. TURÁN: Notes on on-line algorithms, Proc. Conf. SOR (1988), to appear.
- [8] G. GALAMBOS: Parametric lower bound for on-line bin packing, *SIAM J. Alg. Disc. Meth.* 7 (1986), 362—367.
- [9] M. R. GAREY, D. S. JOHNSON: *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.
- [10] R. L. GRAHAM: Bounds for certain multiprocessing anomalies, *Bell Syst. Tech. J.* 45 (1966), 1563—1581.
- [11] A. GYÁRFÁS, J. LEHEL: On-line and first-fit colorings of graphs, *J. Graph Th.* 12 (1988), 217—227.
- [12] D. S. JOHNSON: Approximation algorithms for combinatorial problems, *J. Comp. Syst. Sci.* 9 (1974), 256—278.
- [13] D. S. JOHNSON, A. DEMERS, J. D. ULLMAN, M. R. GAREY, R. L. GRAHAM: Worst-case performance bounds for simple one-dimensional bin packing algorithms, *SIAM J. Comp.* 3 (1974), 299—325.
- [14] H. A. KIERSTEAD: An effective version of Dilworth's theorem, *Trans. AMS* 268 (1981), 63—77.
- [15] H. A. KIERSTEAD: The linearity of first-fit coloring of interval graphs, *SIAM J. Discr. Math.* 1 (1988), 526—530.
- [16] H. A. KIERSTEAD, W. T. TROTTER: An extremal problem in recursive combinatorics, *Congr. Numer.* 33 (1981), 143—153.
- [17] B. KORTE, D. HAUSMANN: An analysis of the greedy heuristic for independence systems, in: *Algorithmic Aspects of Combinatorics* (B. Alspach, P. Hell, D. J. Miller eds.), *Annals of Discr. Math.* 2 (1978), 65—74., North-Holland, Amsterdam.
- [18] E. L. LAWLER: *Combinatorial Optimization: Networks and Matroids*, Holt, Reinhart and Winston, New York, 1976.
- [19] C. C. LEE, D. T. LEE: A simple on-line bin packing algorithm, *J. ACM* 32 (1985), 562—572.
- [20] F. M. LIANG: A lower bound for on-line bin packing, *Inf. Proc. Lett.* 10 (1980), 76—79.
- [21] L. LOVÁSZ: Covers, packings and some heuristic algorithms, Proc. 5th British Comb. Conf., *Util. Math.* (1976), 417—429.
- [22] L. LOVÁSZ: *Combinatorial Problems and Exercises*, Akadémiai Kiadó, Budapest, and North-Holland, Amsterdam, 1979.
- [23] L. LOVÁSZ, M. SAKS, W. T. TROTTER: An on-line graph coloring algorithm with sublinear performance, preprint (1988).
- [24] M. S. MANASSE, L. A. MCGEOCH, D. D. SLEATOR: Competitive algorithms for on-line problems, 20. STOC (1988), 322—333.
- [25] M. SZEGEDY, unpublished (1986).
- [26] E. SZEMERÉDI, unpublished (1982).

- [27] GY. TURÁN: On the greedy algorithm for an edge-partitioning problem, in: Coll. Math. Soc. J. Bolyai 44. Theory of Algorithms, (L. Lovász, E. Szemerédi eds.), Pécs (Hungary), 1984, 405—423.
- [28] W. F. DE LA VEGA, G. S. LUEKER: Bin packing can be solved within $1 + \varepsilon$ in linear time, *Combinatorica* 1 (1981), 349—355.
- [29] V. G. VIZING: On the value of the chromatic class of a p -graph, *Discr. Anal.* 3 (1964), 25—30., Novosibirsk. (In Russian.)
- [30] D. J. A. WELSH: *Matroid Theory*, Academic Press, London, 1976.
- [31] A. C. YAO: New algorithms for bin packing, *J. ACM* 27 (1980), 207—227.

(Received January 5, 1989)